

Тема 11. Проектирование баз данных

1. Жизненный цикл БД
2. Этапы проектирования БД
3. Системный анализ предметной области
4. Инфологическое моделирование предметной области. Модель «сущность-связь»
5. Даталогическое проектирование. Переход от модели «сущность-связь» к реляционной модели. Принципы нормализации

Жизненный цикл баз данных



Этапы проектирования БД

Системный анализ предметной области



Инфологическое проектирование



Выбор СУБД



Даталогическое проектирование



Физическое проектирование

Системный анализ предметной области

- Цель:** провести подробное словесное описание объектов предметной области и реальных связей между объектами
- **Функциональный подход** — реализует принцип движения «от задач», когда заранее известны необходимые функции
 - **Предметный подход** — когда информационные потребности будущих пользователей БД жестко не фиксируются

Системный анализ предметной области

Системный анализ должен включать:

- подробное описание информации об **объектах предметной области**
- формулировку **конкретных задач** с кратким описанием алгоритмов их решения
- **описание выходных документов**, которые должны генерироваться в системе
- **описание входных документов**, которые служат основанием для заполнения данными БД

Пример описания предметной области

Задача: требуется разработать ИС для автоматизации учета получения и выдачи книг в библиотеке

Основные объекты:

- книги и экземпляры книг
- читатели
- выдачи книг на руки

Пример описания предметной области

Параметры, характеризующие каждую книгу:

- уникальный шифр
- название
- фамилии авторов (могут отсутствовать)
- место издания (город)
- издательство
- год издания
- количество страниц
- стоимость книги
- область знаний
- количество экземпляров книги в библиотеке

Пример описания предметной области

На каждого **читателя** в картотеку
вносятся следующие сведения:

- уникальный номер читательского билета
- фамилия, имя, отчество
- домашний адрес
- телефон
- дата рождения

Пример описания предметной области

Каждый **экземпляр книги** имеет:

- уникальный инвентарный номер
- шифр книги, который совпадает с уникальным шифром из описания книг
- место размещения в библиотеке

При **выдаче** экземпляра книги читателю заносятся следующие сведения:

- номер билета читателя, который взял книгу
- дата выдачи книги
- дата возврата

Пример описания предметной области

Предусмотреть следующие **ограничения** :

- Книга может не иметь ни одного автора
- В библиотеке должны быть записаны читатели не моложе 17 лет
- В библиотеке присутствуют книги, изданные начиная с 1960 по текущий год
- Каждый читатель может держать на руках не более 5 книг
- Каждый читатель при регистрации в библиотеке должен дать телефон для связи
- Каждая область знаний может содержать ссылки на множество книг, но каждая книга может относиться к различным областям знаний

Пример описания предметной области

С данной ИС должны работать следующие **группы пользователей**:

- библиотекари
- читатели
- администрация библиотеки

Затем необходимо определить, какие **задачи** будет решать каждый пользователь (или группа пользователей)

Инфологическое моделирование

- Инфологическое проектирование связано с представлением **семантики** предметной области в модели базы данных
- Инфологическое описание не должно быть привязано к конкретной СУБД
- Инфологическая (семантическая) модель представляет собой емкое формализованное описание предметной области

Модель «сущность-связь»

Модель «сущность-связь»

(Entity-Relationship model, ER-модель)

- ER-модель является **концептуальной моделью**, т.е. не учитывает особенности конкретной СУБД
- Из модели могут быть получены все основные фактографические модели данных
- Процесс создания модели является **итерационным** (уточняющим)

Модель «сущность-связь»: ПОНЯТИЯ

В основе ER-модели лежат следующие базовые понятия:

- **Сущности**
- **Атрибуты**
- **Связи**

Модель «сущность-связь»:

сущность

Сущность — это реальный или представляемый объект, информация о котором должна сохраняться в проектируемой системе

- Сущность имеет имя, **уникальное** в пределах системы
- Сущность соответствует некоторому классу однотипных объектов (существует множество экземпляров данной сущности)

Модель «сущность-связь»: атрибуты

- Объект имеет свой набор атрибутов — характеристик, определяющих свойства данного объекта
- Атрибут должен иметь имя, **уникальное** в пределах данной сущности
- Ключ сущности — это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности

Модель «сущность-связзь»: сущность

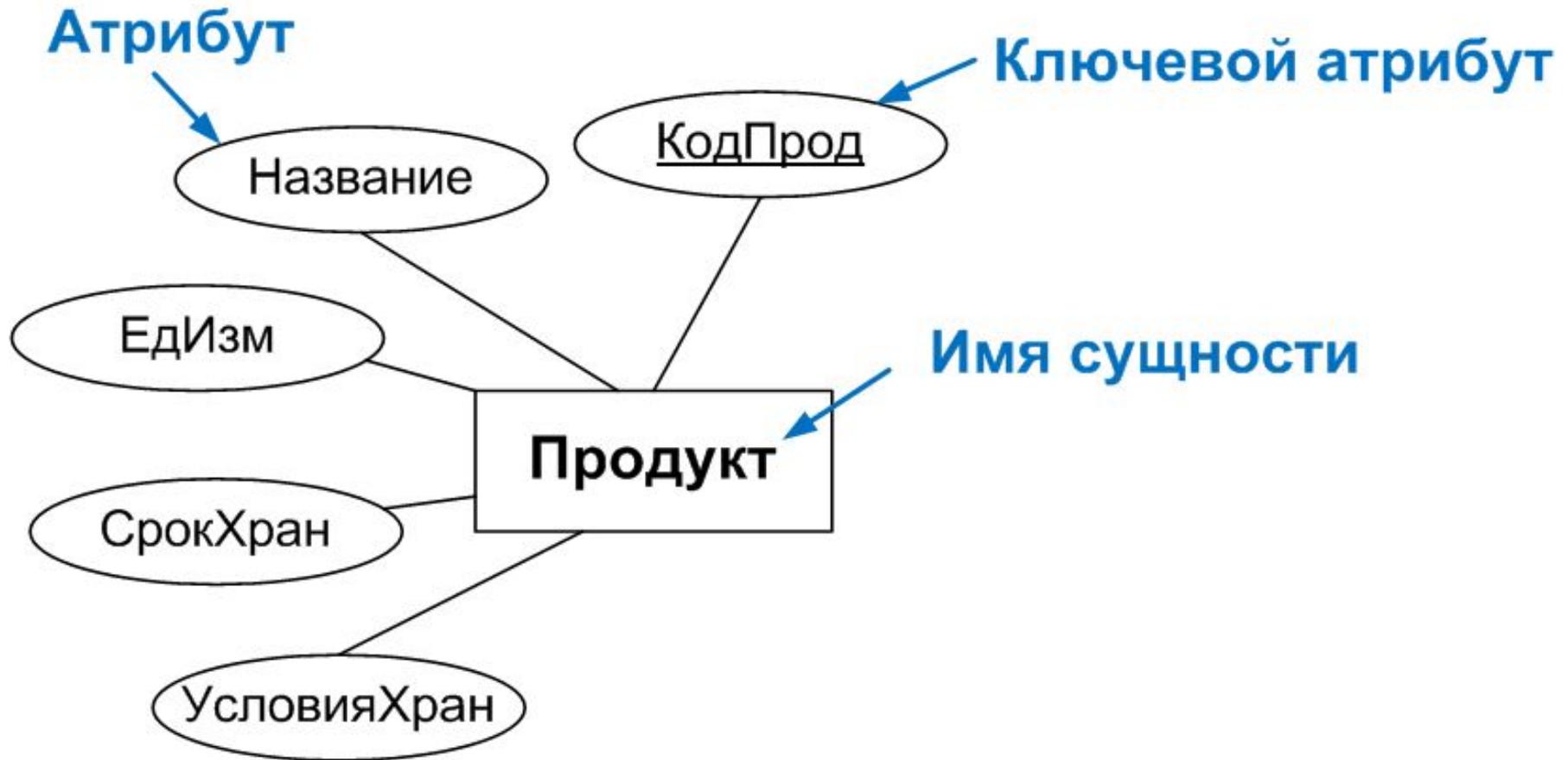
Имя сущности



Ключевой атрибут

Атрибуты

Модель «сущность-связь»: сущность



Модель «сущность-связь»: СВЯЗЬ

Связь — это ассоциация, установленная между несколькими сущностями и показывающая, как взаимодействуют сущности между собой

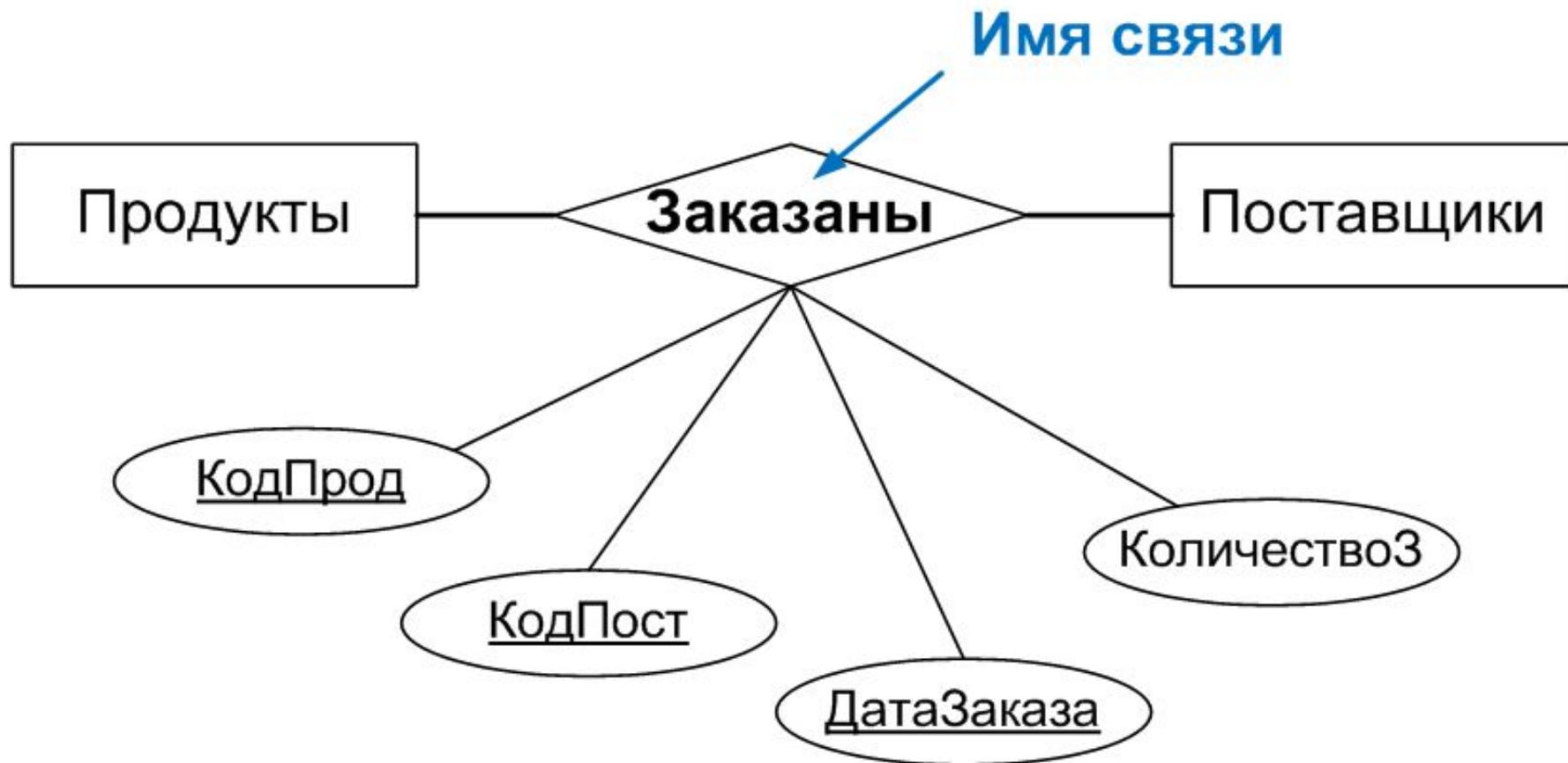
- Связь определяет взаимосвязь между **экземплярами сущностей**
- Связь также может иметь **атрибуты**
- Между сущностями может быть задано сколько угодно связей с различными смысловыми нагрузками

Модель «сущность-связь»: СВЯЗЬ

Связь может существовать:

- между двумя разными сущностями
(**бинарная связь**)
- между n сущностями (**n -арная связь**)
- между сущностью и ей же самой
(**рекурсивная связь**)

Модель «сущность-связь»: СВЯЗЬ



Модель «сущность-связь»: СВЯЗЬ

Степень связи — число экземпляров сущностей, которое может быть ассоциировано через связь с экземплярами другой сущности

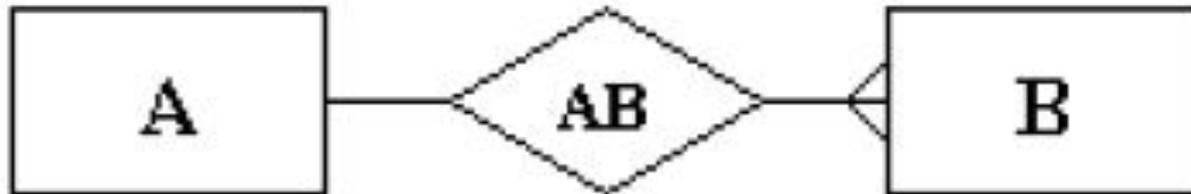
Модель «сущность-связь»: СВЯЗЬ

Степени бинарных связей:

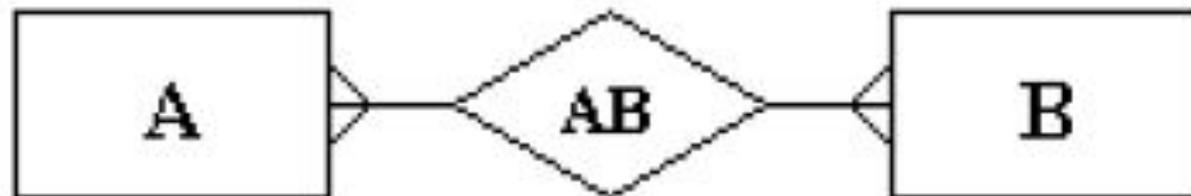
- ОДИН-К-ОДНОМУ (1:1)



- ОДИН-КО-МНОГИМ (1:M)



- МНОГИЕ-КО-МНОГИМ (M:N)



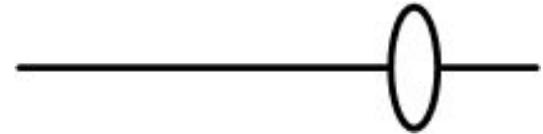
Модель «сущность-связь»: СВЯЗЬ

Класс принадлежности входящих в связь сущностей:

- Связь любого из типов может быть обязательной, если в данной связи должен участвовать каждый экземпляр сущности
- Связь любого из типов может быть необязательной, если не каждый экземпляр сущности должен участвовать в данной связи

Модель «сущность-связь»: СВЯЗЬ

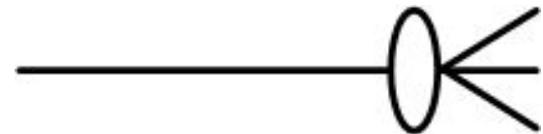
- Связь степени 1, необязательный класс



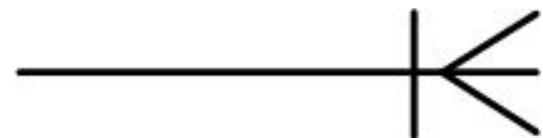
- Связь степени 1, обязательный класс



- Связь степени N, необязательный класс



- Связь степени N, обязательный класс



Модель «сущность-связь»: примеры

Примеры связей один-к-одному:



Модель «сущность-связь»: примеры

Примеры связей один-ко-многим:



Модель «сущность-связь»: СВЯЗЬ

Если существование сущности x зависит от существования сущности y , то x называется зависимой сущностью



Модель «сущность-связь»: примеры

Примеры связей многие-ко-многим:



Между одними и теми же сущностями могут существовать **несколько** связей:



Модель «сущность-связь»: построение

Этапы построения диаграммы «сущность-связь»:

1. Определение списка сущностей выбранной предметной области
2. Определение списка атрибутов сущностей
3. Описание связей между сущностями (степени, классы принадлежности связей, а также атрибуты связей, если они необходимы)
4. Организация данных в виде диаграммы "сущность-связь"

Модель «сущность-связь»: пример

Задача: построить диаграмму, отображающую связь данных для информационной системы учета продажи продуктов в магазине.

БД должна хранить информацию:

- о **продуктах**, поставляемых в магазин
- об ежедневной **продаже** продуктов
- о **заказах на поставку** продуктов
- о **поставщиках** продуктов

Модель «сущность-связь»:

пример

Составим список сущностей с их атрибутами:

1. Сущность «**Продукты**»

- *Код продукта* – уникальный идентификатор, ключевой атрибут
- *Продукт* – название продукта
- *Единица измерения* – литры, килограммы, штуки и т.п.
- *Срок хранения в днях* – для определения даты окончания срока годности продукта
- *Условия хранения* – температура, влажность и т.п.

Модель «сущность-связь»:

пример

2. Сущность «Поставщики»

- *Код поставщика* – уникальный идентификатор, ключевой атрибут
- *Поставщик* – название организации или ФИО физического лица
- *Код города* – город, где находится поставщик (для поиска)
- *Адрес* – улица и дом (а также квартира – для физического лица)
- *ФИО директора*
- *Телефон*
- *Факс*

Модель «сущность-связь»:

пример

3. Сущность «Продажи»

- Дата продажи
- Код продукта – какой именно продукт был продан
- *Количество* – сколько продано этого продукта в тех единицах измерения, которые указаны для этого продукта в сущности
Продукт
- *Цена продажи* – цена при продаже за единицу продукта

Модель «сущность-связь»: пример

4. Сущность «Города»

- Код города – уникальный идентификатор, ключевой атрибут
- *Город* – название города

Модель «сущность-связь»: пример

Рассмотрим связи, существующие между сущностями:

1. Связь M:N «**Поставляют**» между сущностями **Продукты** и **Поставщики**



Модель «сущность-связь»: пример

Связь «**Поставляют**» имеет следующие атрибуты:

- Дата поставки
- Код поставщика – какой поставщик поставил этот продукт
- Код продукта – какой именно продукт был поставлен
- *КоличествоП* – сколько поставлено этого продукта
- *Цена поставки* – цена при поставке за единицу продукта
- *Дата изготовления* – дата изготовления продукта

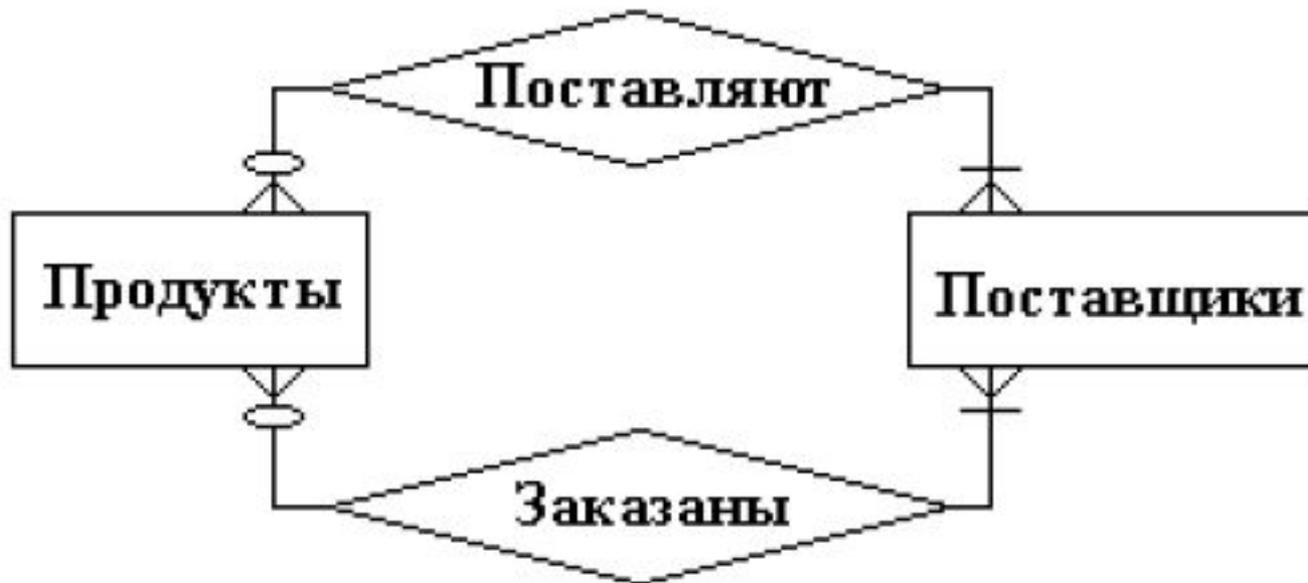
Модель «сущность-связь»: пример

2. Связь M:N «**Заказаны**» между сущностями Продукты и Поставщики

- Дата заказа
- Код поставщика – какому поставщику заказан этот продукт
- Код продукта – какой именно продукт был заказан
- *Количество*3 – сколько поставлено этого продукта

Модель «сущность-связь»: пример

Связи между сущностями Продукты и
Поставщики:

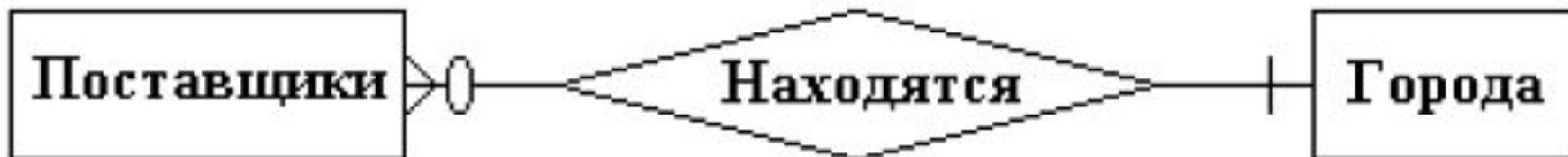


Модель «сущность-связь»: пример

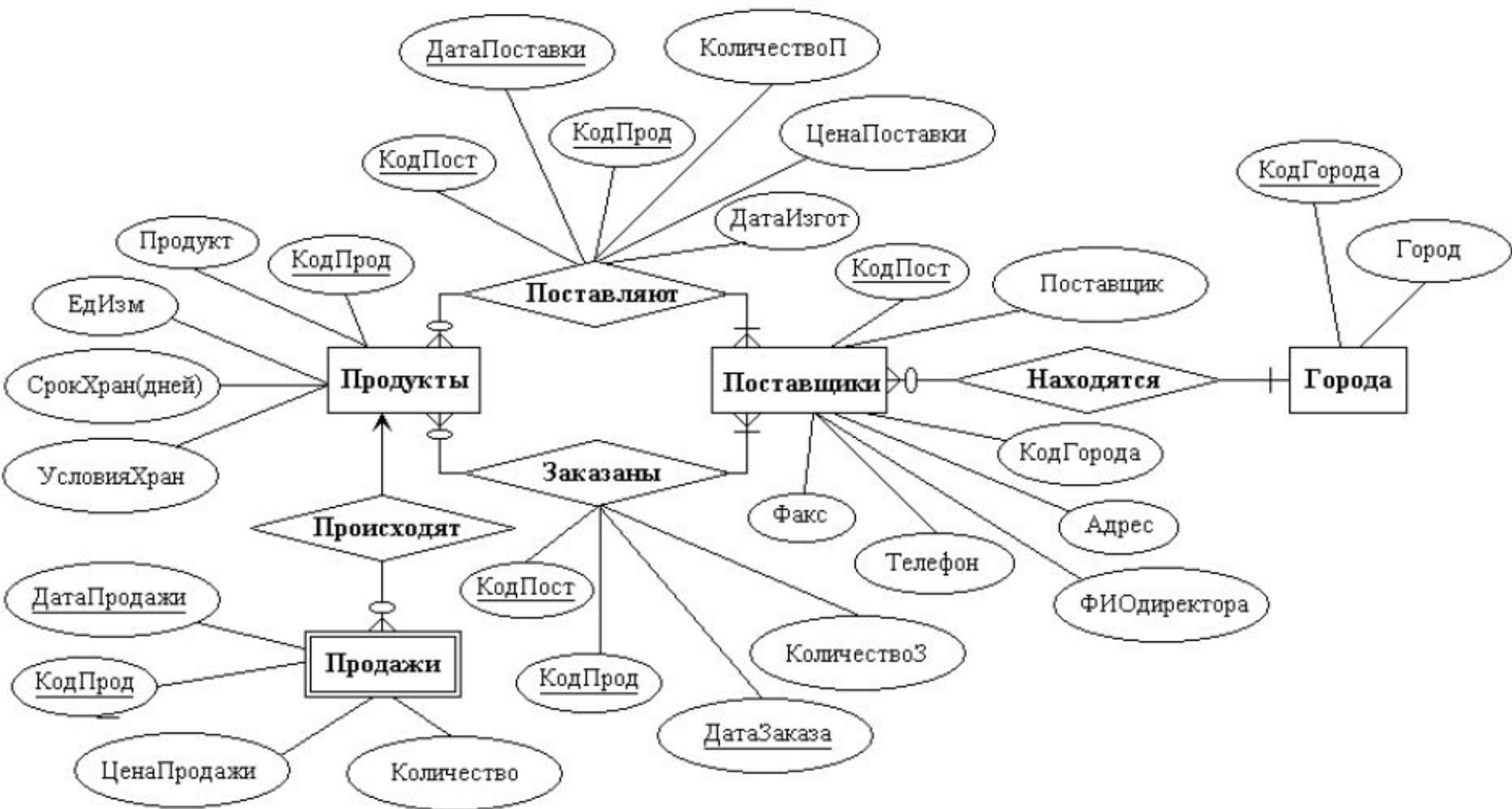
3. Связь N:1 «**Происходят**» между сущностями Продажи и Продукты



4. Связь N:1 «**Находятся**» между сущностями Поставщики и Города



Модель «сущность-связь»: пример



Инфологическое моделирование: CASE

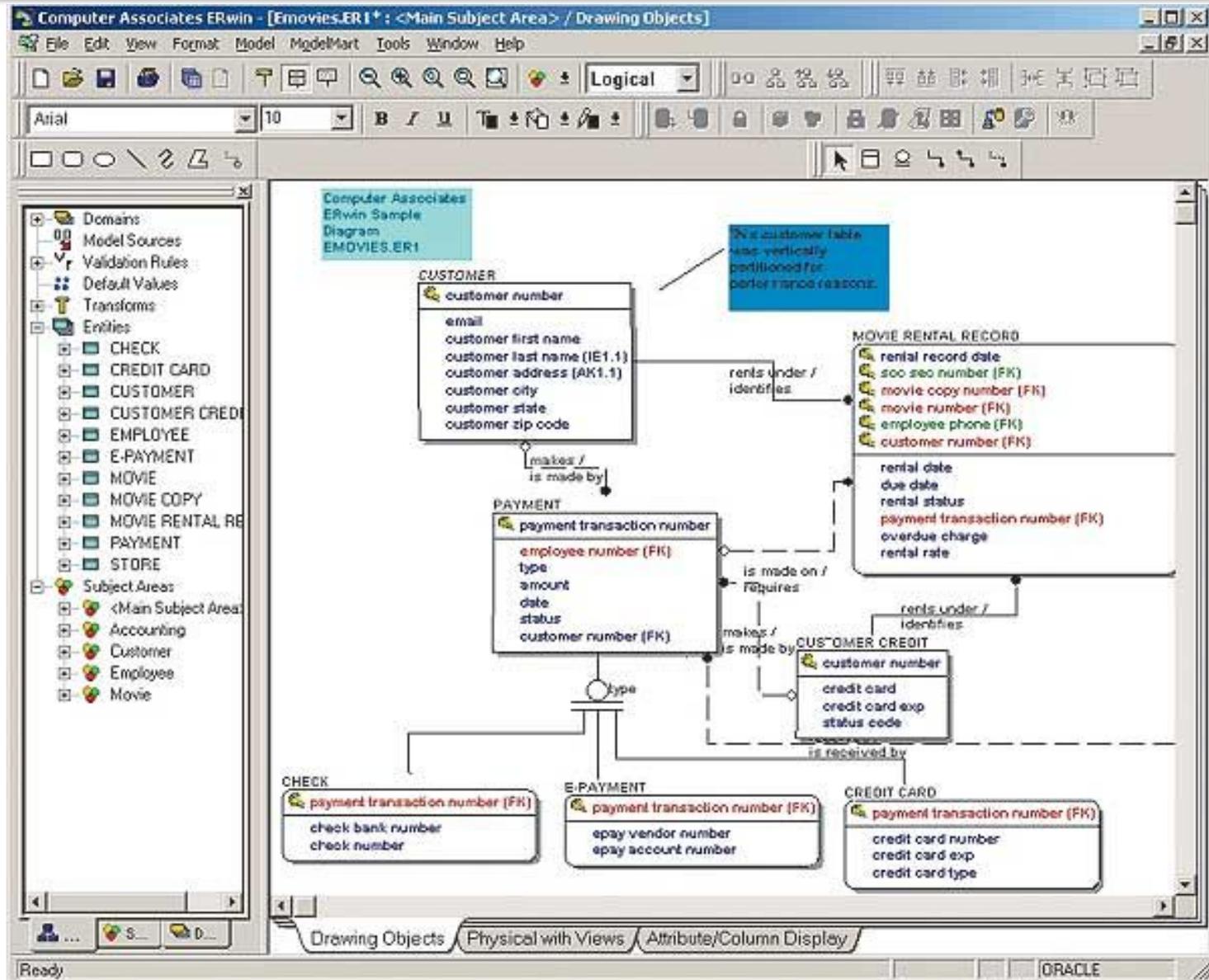
CASE-средства

Computer-Aided System (Software) Engineering

CASE-средства обеспечивают поддержку технологий автоматизированного проектирования, разработки и сопровождения программных систем

Пример: AllFusion ERwin Data Modeler
(ERwin)

Инфологическое моделирование: CASE



Алгоритм перехода к реляционной модели

1. Каждой сущности модели «сущность-связь» ставится в соответствие отношение реляционной модели
2. Каждый атрибут сущности становится атрибутом соответствующего отношения:
 - задается конкретный допустимый в СУБД тип данных
 - обязательность или необязательность данного атрибута (допустимость или недопустимость NULL-значений)

Алгоритм перехода к реляционной модели

3. Первичный ключ сущности становится первичным ключом соответствующего отношения
4. В каждое отношение, соответствующее сущности со стороны «многие» (связь **1:M**), добавляется набор атрибутов сущности со стороны «один», являющихся первичным ключом сущности со стороны «один»



Алгоритм перехода к реляционной модели

5. Для моделирования необязательного и обязательного класса принадлежности:
- у атрибутов сущности **необязательного класса** принадлежности, соответствующих **внешнему ключу**, устанавливается свойство **допустимости неопределенных значений**
 - при **обязательном классе** принадлежности атрибуты получают свойство **отсутствия неопределенных значений**

Алгоритм перехода к реляционной модели

6. Разрешение связей типа M:N:

- Связи становятся в соответствие новое отношение, имеющее атрибуты, которые в сущностях являются **первичными ключами**, а в новом отношении будут **внешними ключами**
- Первичным ключом нового отношения будет **совокупность внешних ключей**



Пример перехода к реляционной модели

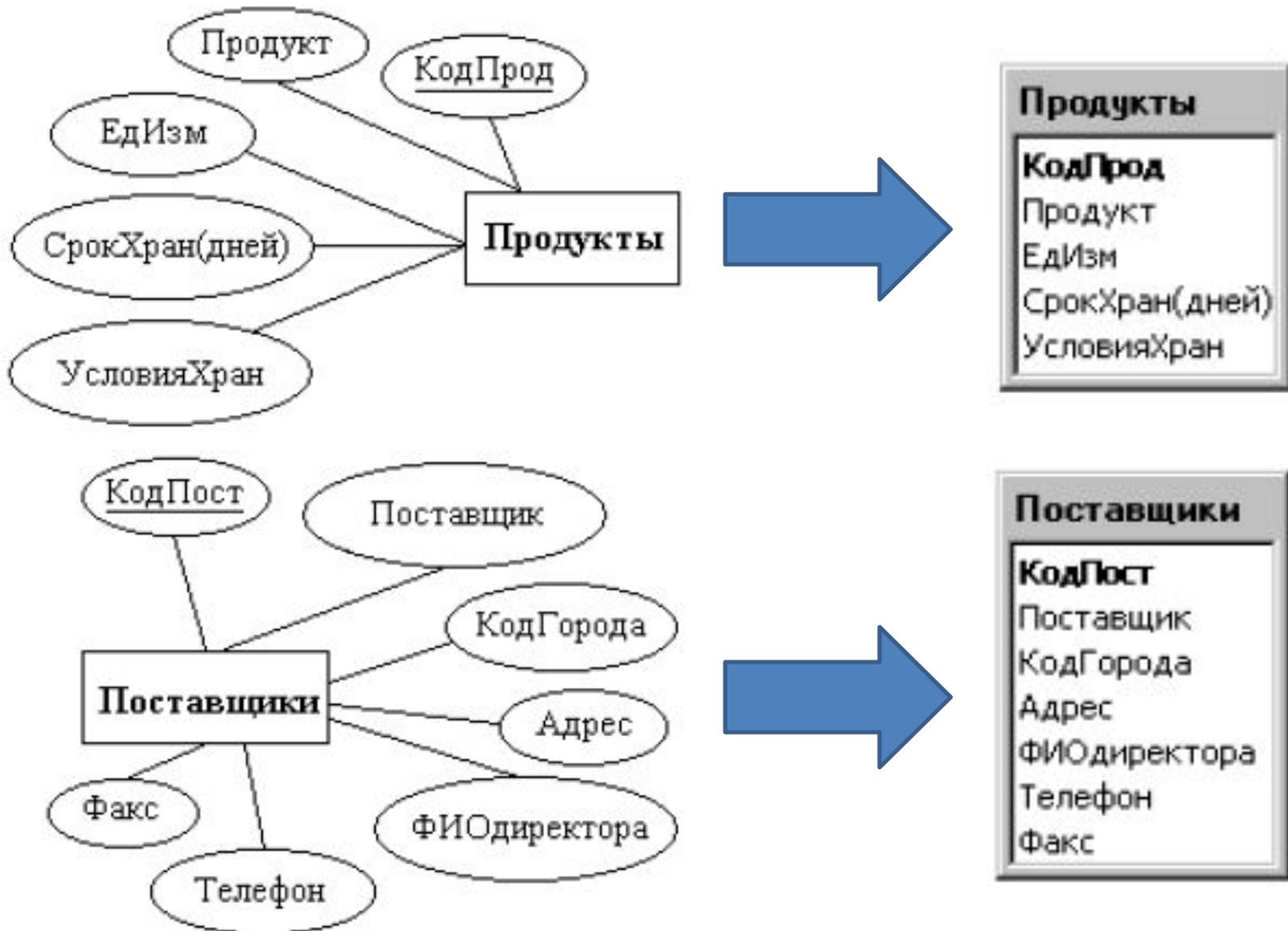
Пример преобразования модели «сущность-связь» к реляционной модели:

В указанной модели мы имеем дело со следующими сущностями:

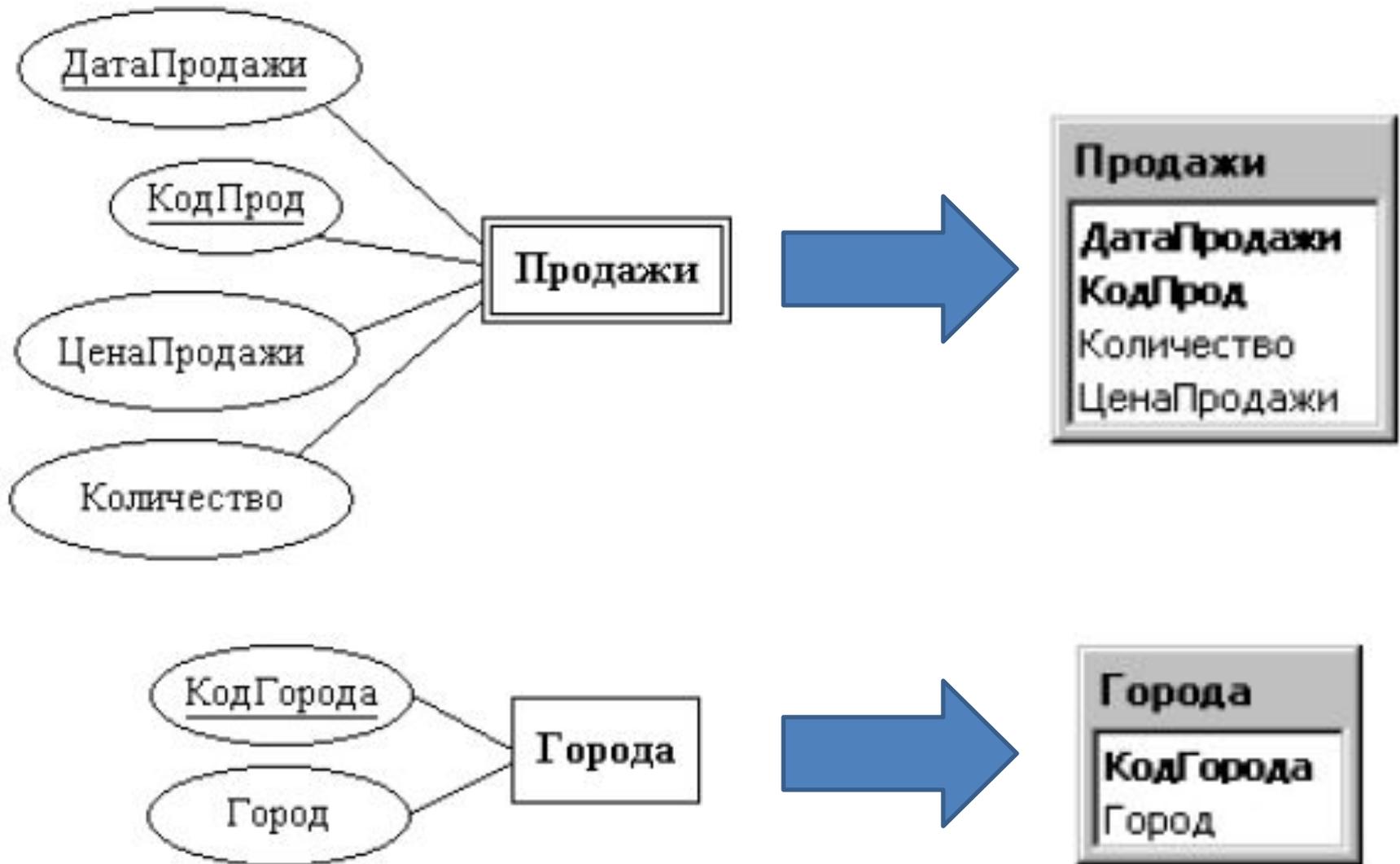
- Продукты
- Поставщики
- Города
- Продажи

Следовательно, и в реляционной модели будут участвовать четыре отношения с такими же именами.

Пример перехода к реляционной модели



Пример перехода к реляционной модели



Пример перехода к реляционной модели

Схема отношения «Продукты»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>КодПрод</i>	Целое	Да	+	
<i>Продукт</i>	Текстовый (30)	Да		
<i>ЕдИзм</i>	Текстовый (5)	Нет		
<i>СрокХран</i>	Целое	Нет		
<i>УсловияХран</i>	Текстовый (200)	Нет		

Пример перехода к реляционной модели

Схема отношения «Поставщики»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>КодПост</i>	Целое	Да	+	
<i>Поставщик</i>	Текстовый (50)	Да		
<i>КодГорода</i>	Целое	Да		+
<i>Адрес</i>	Текстовый (100)	Нет		
<i>ФИОдиректора</i>	Текстовый (50)	Нет		
<i>Телефон</i>	Текстовый (15)	Нет		
<i>Факс</i>	Текстовый (15)	Нет		

Пример перехода к реляционной модели

Схема отношения «Продажи»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>ДатаПродажи</i>	Дата/время	Да	+	
<i>КодПрод</i>	Целое	Да		+
<i>Количество</i>	Одинарное с плавающей точкой	Нет		
<i>ЦенаПродажи</i>	Денежный	Нет		

Пример перехода к реляционной модели

Схема отношения «Города»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>КодГорода</i>	Целое	Да	+	
<i>Город</i>	Текстовый (30)	Да		

Пример перехода к реляционной модели

В примере две связи имеют степень M:N.
Это связи **Поставляют** и **Заказаны**.

Следовательно, дополнительно
появляются еще два отношения:

- **Поставки**
- **Заказы**

Пример перехода к реляционной модели

Схема отношения «Поставки»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>ДатаПоставки</i>	Дата/Время	Да	+	
<i>КодПост</i>	Целое	Да		+
<i>КодПрод</i>	Целое	Да		+
<i>КоличествоП</i>	Одинарное с плавающей точкой	Нет		
<i>ЦенаПоставки</i>	Денежный	Нет		
<i>ДатаИзгот</i>	Дата время	Нет		

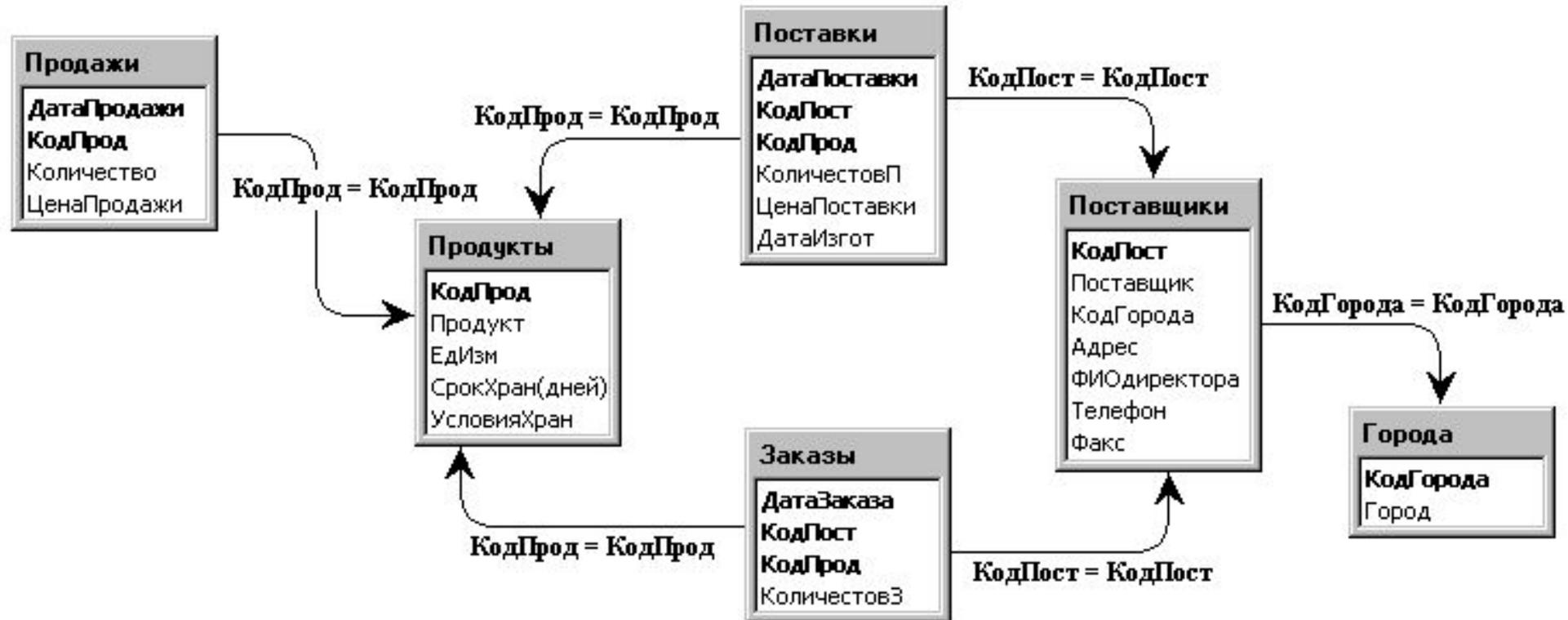
Пример перехода к реляционной модели

Схема отношения «Заказы»

Атрибут	Тип данных (СУБД Access)	Обязательный атрибут	Первичный ключ	Внешний ключ
<i>ДатаЗаказа</i>	Дата/Время	Да	+	
<i>КодПост</i>	Целое	Да		+
<i>КодПрод</i>	Целое	Да		+
<i>КоличествоЗ</i>	Одинарное с плавающей точкой	Нет		

Пример перехода к реляционной модели

Окончательный вариант реляционной модели (Схемы БД)



Даталогическое проектирование

Цель даталогического проектирования:

разработка корректной **схемы БД** в терминах выбранной СУБД

Основой анализа корректности схемы являются анализ **функциональных зависимостей** между атрибутами отношений БД

Даталогическое проектирование

Инфологическое
проектирование

Модель
«сущность-связь»

Переход к реляционной модели

Даталогическое
проектирование

Ненормализованная
схема БД

Анализ функциональных зависимостей

Нормализация схемы БД

Нормализованная
схема БД

Даталогическое проектирование

После нормализации схемы БД и окончательного выбора СУБД выполняется:

- Описание **концептуальной схемы БД** в терминах выбранной СУБД
- Описание **внешних моделей** в терминах выбранной СУБД
- Описание правил поддержки **целостности** базы данных
- Разработка процедур поддержки **семантической целостности** базы данных

Проектирование схемы БД

Проектирование схемы БД может быть выполнено двумя путями:

- **путем декомпозиции (разбиения):**
путем последовательной **нормализации** схем отношений
- путем синтеза

Универсальное отношение — это таблица, в которую включены все интересующие атрибуты, то есть та таблица, которая требует нормализации

Нормализация базы данных

Нормализация — это процесс преобразования отношения в состояние, обеспечивающее лучшие условия выборки, добавления, изменения и удаления данных.

Главная цель нормализации: устранение избыточности и дублирования информации в базе данных

Нормальные формы

первая нормальная форма (1NF)



вторая нормальная форма (2NF)



третья нормальная форма (3NF)



нормальная форма Бойса—Кодда
(BCNF)



четвертая нормальная форма (4NF)



пятая нормальная форма (5NF)

Свойства нормальных форм

Каждой нормальной форме соответствует определенный набор ограничений

Основные свойства нормальных форм:

- каждая следующая нормальная форма улучшает свойства предыдущей
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм **сохраняются**

Первая нормальная форма

Отношение находится в **первой нормальной форме**, если значения всех его атрибутов **атомарны**.

R1 - Ненормализованное отношение

КодПоставщика	КодПродукта	Продукт
P ₁	1	Сахар
	2	Соль
	13	Мука
P ₂	26	Рис
	58	Гречка
	130	Крупа манная
	162	Пшено
P ₃	474	Молоко
	891	Кефир

R2 - Нормализованное отношение

КодПоставщика	КодПродукта	Продукт
P ₁	1	Сахар
P ₁	2	Соль
P ₁	13	Мука
P ₂	26	Рис
P ₂	58	Гречка
P ₂	130	Крупа манная
P ₂	162	Пшено
P ₃	474	Молоко
P ₃	891	Кефир

Первая нормальная форма: пример

Преподаватель	День недели	Номер пары	Название дисциплины	Тип занятий	Группа
Петров В. И.	Понедельник	1	Теор. выч. проц.	Лекция	4906
	Вторник	1	Комп. графика	Лаб. раб.	4907
	Вторник	2	Комп. графика	Лаб. раб.	4906
Киров В. А.	Понедельник	2	Теория информ.	Лекция	4906
	Вторник	3	Пр-е на C++	Лаб. раб.	4907
	Вторник	4	Пр-е на C++	Лаб. раб.	4906
Серов А. А.	Понедельник	3	Защита инф.	Лекция	4944
	Среда	3	Базы данных	Лаб. раб.	4942
	Четверг	4	Базы данных	Лаб. раб.	4922

Первая нормальная форма: пример

Преподаватель	День недели	Номер пары	Название дисциплины	Тип занятий	Группа
Петров В. И.	Понедельник	1	Теор. выч. проц.	Лекция	4906
Петров В. И.	Вторник	1	Комп. графика	Лаб. раб.	4907
Петров В. И.	Вторник	2	Комп. графика	Лаб. раб.	4906
Киров В. А.	Понедельник	2	Теория информ.	Лекция	4906
Киров В. А.	Вторник	3	Пр-е на С++	Лаб. раб.	4907
Киров В. А.	Вторник	4	Пр-е на С++	Лаб. раб.	4906
Серов А. А.	Понедельник	3	Защита инф.	Лекция	4944
Серов А. А.	Среда	3	Базы данных	Лаб. раб.	4942
Серов А. А.	Четверг	4	Базы данных	Лаб. раб.	4922

Недостатки первой нормальной формы

- **избыточность** — многократное повторение информации в столбцах данных
- аномалии модификации (обновления) данных
- аномалии добавления данных
- аномалии удаления данных

Пример:

Экзамены (ФИО, Номер зач.кн., Группа, Дисциплина, Дата экзамена, Оценка)

Избыточность данных: пример

ФИО	Номер ЗачКн	Группа	Название дисциплины	Дата	Оценка
Пупкин В. И.	323556	ММ-117	Управление данными	17/01/10	2
Пупкин В. И.	323556	ММ-117	Управление данными	25/01/10	3
Петров В. А.	156900	ММ-117	Управление данными	25/01/10	5
Сидоров А. А.	278001	ММ-119	Мат. анализ	21/01/10	5
Киров В. У.	777890	ММ-119	Мат. анализ	21/01/10	4
Хренова Г. П.	123456	ММ-334	Инф. менеджмент	21/01/10	3
Бобриков С. С.	998769	ММ-334	Инф. менеджмент	21/01/10	5
Хренова Г. П.	123456	ММ-334	Базы данных	24/01/10	2
Бобриков С. С.	998769	ММ-334	Базы данных	24/01/10	4 ⁷⁰

Функциональная зависимость

Атрибут Y некоторого отношения функционально зависит от X (атрибуты могут быть составными), если в любой момент времени каждому значению X соответствует одно значение Y .

Функциональная зависимость

обозначается: $X \twoheadrightarrow Y$

Пример: Номер зач.кн. \twoheadrightarrow ФИО

Полная функциональная зависимость

Неключевой атрибут **функционально полно** зависит от **составного ключа**, если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

Пример:

Номер зач.кн., Дисциплина, Дата ►
Оценка

Вторая нормальная форма

Отношение (таблица) находится во 2НФ, если оно находится в 1НФ, и **каждый неключевой атрибут функционально полно** зависит от всего ключа.

Приводить ко 2 НФ необходимо только отношения с составным ключом

Вторая нормальная форма

Если какой-либо атрибут зависит от **части составного первичного ключа**, то необходимо:

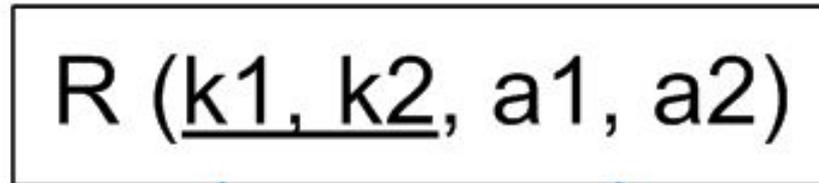
- создать новое отношение, атрибутами которого будут:
- часть составного ключа (первичный ключ нового отношения)
- атрибут, зависящий от нового ключа
- из исходного отношения исключить атрибут, включенный в новое отношение

Вторая нормальная форма

Полная ФЗ: $\underline{k1, k2} \rightarrow a1$

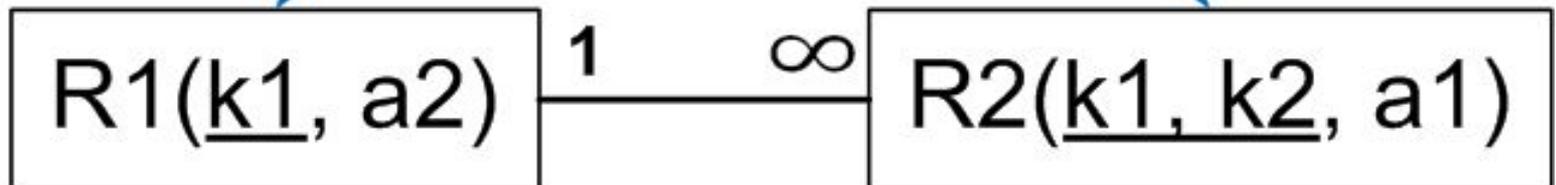
Неполная ФЗ: $\underline{k1} \rightarrow a2$

1 НФ



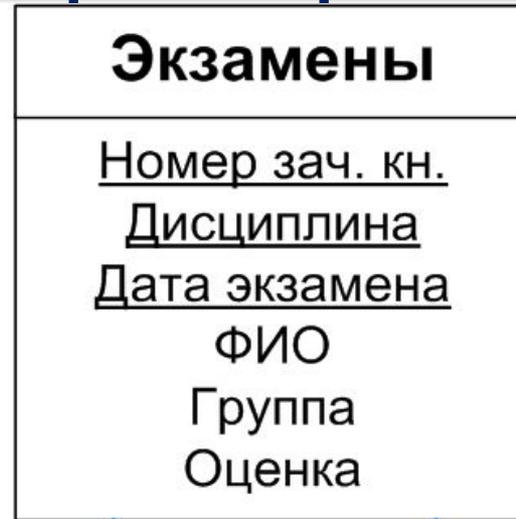
Проекция

2 НФ



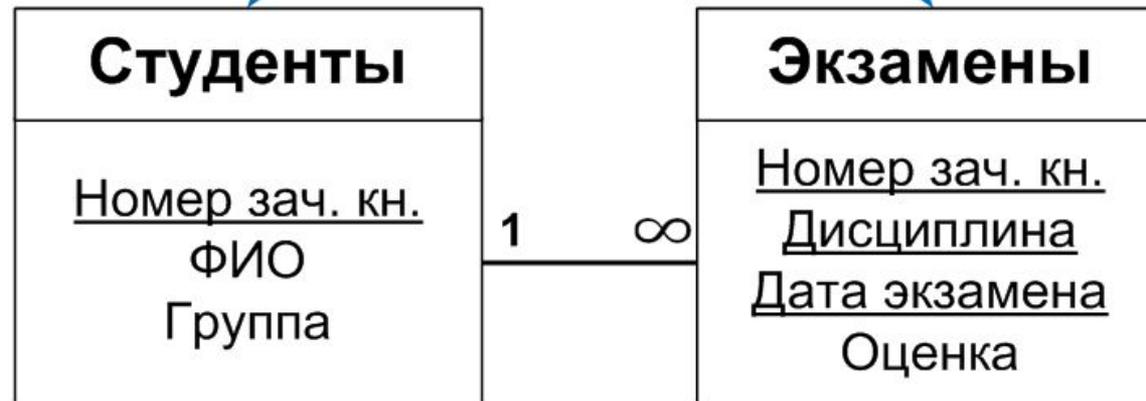
Вторая нормальная форма: пример

1 НФ



Проекция

2 НФ



Определение неполных ФЗ

Составление таблицы-опросника:

КЛ – ключевые атрибуты, НК – неключевые

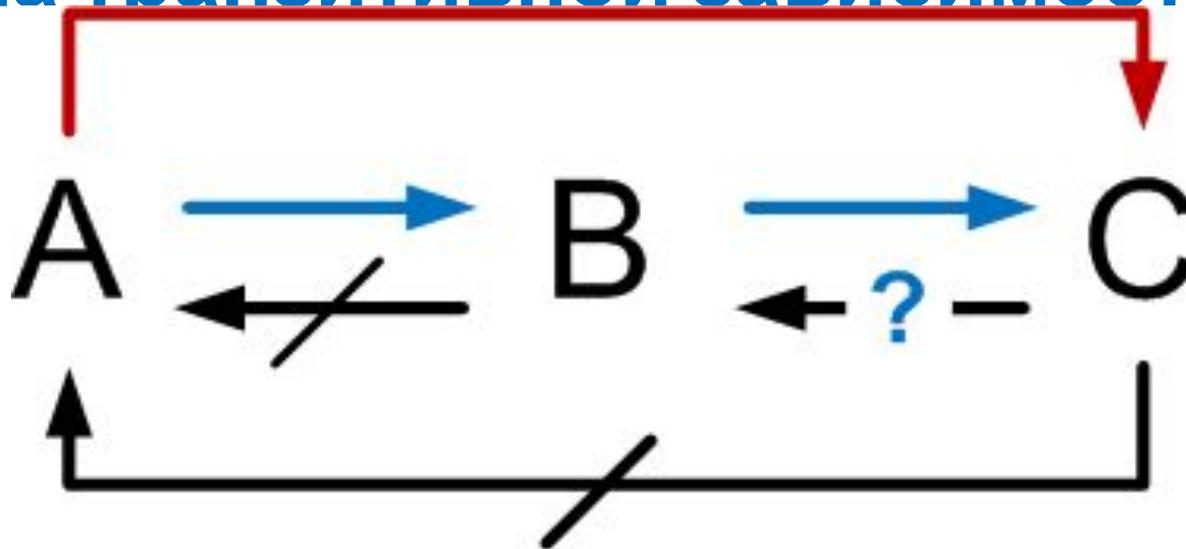
КЛ	КЛ1	КЛ2	...	КЛn
НК				
НК1	+	+	+	+
НК2		+		+
...	+	+	+	+
НКn	+			

Транзитивная зависимость

Транзитивная функциональная зависимость:

Пусть A, B, C – три атрибута некоторого отношения R.

Схема транзитивной зависимости:



Третья нормальная форма

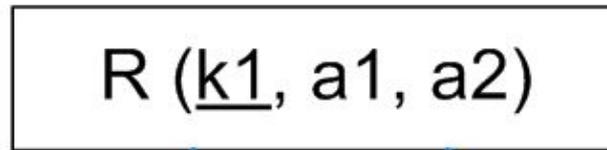
Отношение находится в 3НФ, если оно находится во 2НФ и **каждый неключевой атрибут нетранзитивно** зависит от первичного ключа.

Наличие транзитивной зависимости влечет за собой появление **аномалий обновления**.

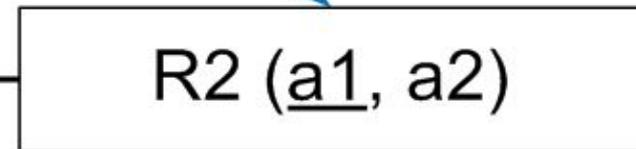
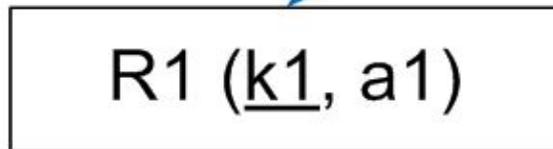
Третья нормальная форма

Транзитивная зависимость:
 $k1 \rightarrow a1$ ($a1 \not\rightarrow k1$)
 $a1 \rightarrow a2$ ($a2 \not\rightarrow a1$)

2 НФ



Проекция



∞

1

3 НФ

Третья нормальная форма: пример

**Транзитивная
зависимость:**

Группа → Специальность

Специальность → Факультет

2 НФ



Проекция

3 НФ



∞

1



Определение транзитивных ФЗ

Составление таблицы-опросника:

НК – неключевые атрибуты

НК	НК1	НК2	...	НКn
НК1		+		
НК2	+			
...				
НКn		+		