

# Базы Данных

2009

# Общее представление о БД

**База Данных (БД)** — структурированный организованный набор данных, описывающих характеристики каких-либо физических или виртуальных систем.

Предметная область

Сущность

Атрибут

Первичный ключ (простой или составной)

Ключ доступа

Ключевой атрибут

Неключевой атрибут

Внешний ключ

# Общее представление о БД

Пример 1.1. Предметная область ВУЗ. Сущность  
Студент, атрибуты НОМЕР\_СТУДЕНТА, ФИО,  
ГРУППА, АДРЕС

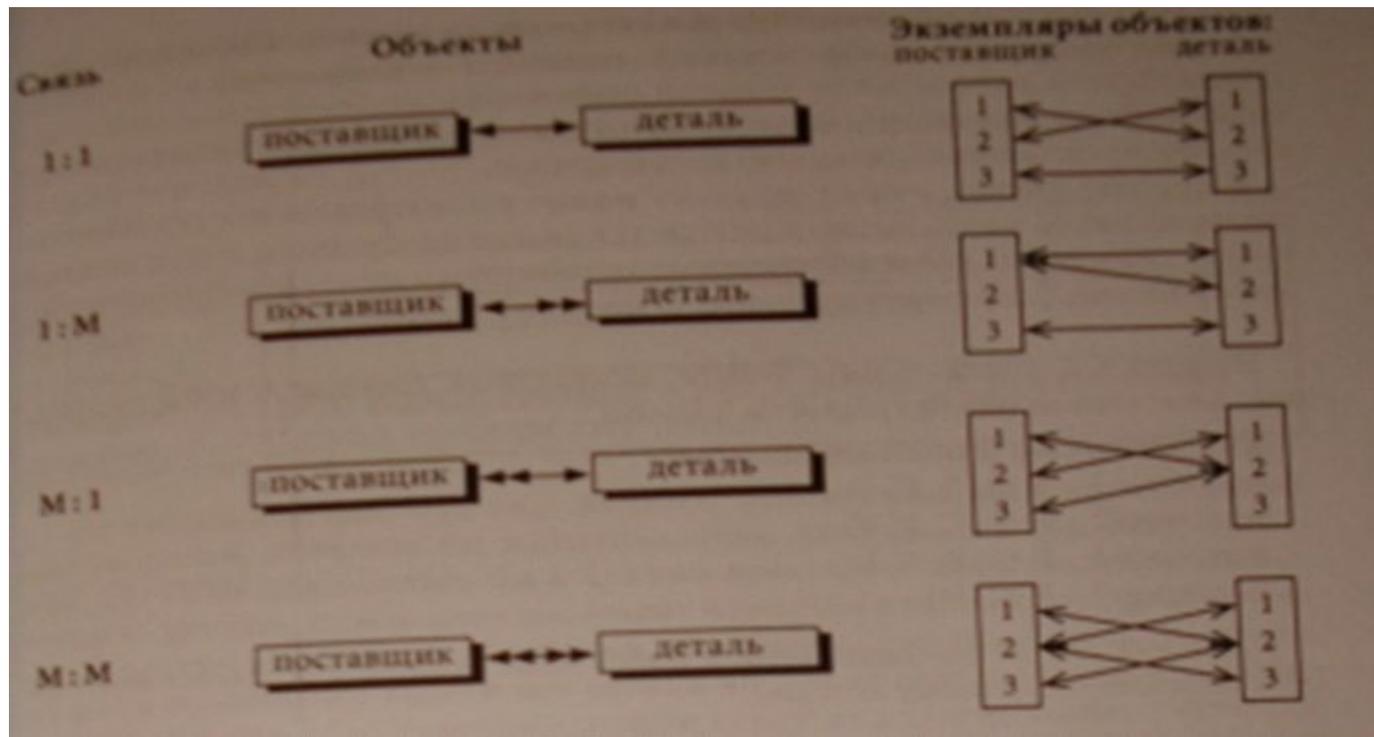
- 1 Иванов И.В. 478 Институтский пер.
- 2 Петров А.Ю. 789 Суворовский пер.
- 3 Иванов И.В. 278 Институтский пер.
- 4 Сидоров А.К. 789 Институтский пер.

.....

# Общее представление о БД

Связи – ассоциации, описывающие взаимоотношения между сущностями предметной области.

- ИМЯ
- полнота (возможная, обязательная, условная ...)
- однозначность



# Уровни представления данных в информационной системе

Локальное представление  
пользователя А

Концептуальное пред.  Физическое пред.

Локальное представление  
пользователя В

Внешний

| Глобальный

Логический уровень

| Физический уровень

# Уровни представления данных в информационной системе

**Пример 1.2.** ВУЗ, сущность Студент. Декан, которого интересует успеваемость. Ректорат, который может оказать материальную помощь студенту.

Пользователь	Атрибуты, интересующие пользователя
Декан	<u>ФИО</u> , <u>НОМЕР_ГРУППЫ</u> , УСПЕВАЕМОСТЬ
Ректорат	<u>ФИО</u> , <u>НОМЕР_ГРУППЫ</u> , АДРЕС, СЕМЕЙНОЕ ПОЛОЖЕНИЕ, ДОХОД

Интеграция локального представления в единое концептуальное: Схема=подсхема1 U подсхема2= {ФИО, Номер\_группы, успеваемость, адрес, семейное положение, доход}

# Интегрированная база данных

- Сокращение избыточности данных;
- Централизованное управление со стороны администратора базы данных;
- Устранение(до некоторой степени) противоречивости данных;
- Поддержание данных в актуальном состоянии;
- Общий доступ к данным из нескольких приложений;
- Соблюдение стандартов;
- Обеспечение целостности данных;
- Наращивание данных для новых приложений пользователей;
- Независимость физического хранения данных от приложений пользователей.

# Основные требования, предъявляемые к базам данных

1. Минимальная избыточность(полезная и вредная)
2. Целостность данных («правильность» и актуальности данных)
3. Безопасность и секретность
4. Независимость данных
5. Производительность
6. Гибкость и способность к расширению

## 2. Проблемы проектирования интегрированных баз данных.

### 2.1 Этапы проектирования.



Рис. 2.1. Этапы проектирования интегрированной базы данных

## 2. Проблемы проектирования интегрированных БД.

### 2.2 Проектирование СУБД-независимого концептуального представления данных.

- Выбрать сущности предметной области
- Описать локальные представления пользователей
- Выполнить интеграцию локальных представлений пользователей
- Окончательно определить сущности и атрибуты

Сложности:

-выделение сущностей. «Назначение» - как связь в  
ПРОЕКТ ---СЛУЖАЩИЙ

Как сущность НАЗНАЧЕНИЕ(ПРОЕКТ,СЛУЖАЩИЙ)

Как атрибут   ПРОЕКТ(НАЗНАЧЕННЫЕ СЛУЖАЩИЕ)  
                  СЛУЖАЩИЙ( НАЗНАЧЕННЫЙ\_НА\_ПРОЕКТ)

## 2. Проблемы проектирования интегрированных БД.

### 2.2 Проектирование СУБД-независимого концептуального представления данных.

#### **Рекомендации:**

- Использовать ту конструкцию, которая кажется более естественной
- Избегать избыточности в использовании конструкции
- 7+/-2
- Каждой сущности точное, а не расплывчатое именование

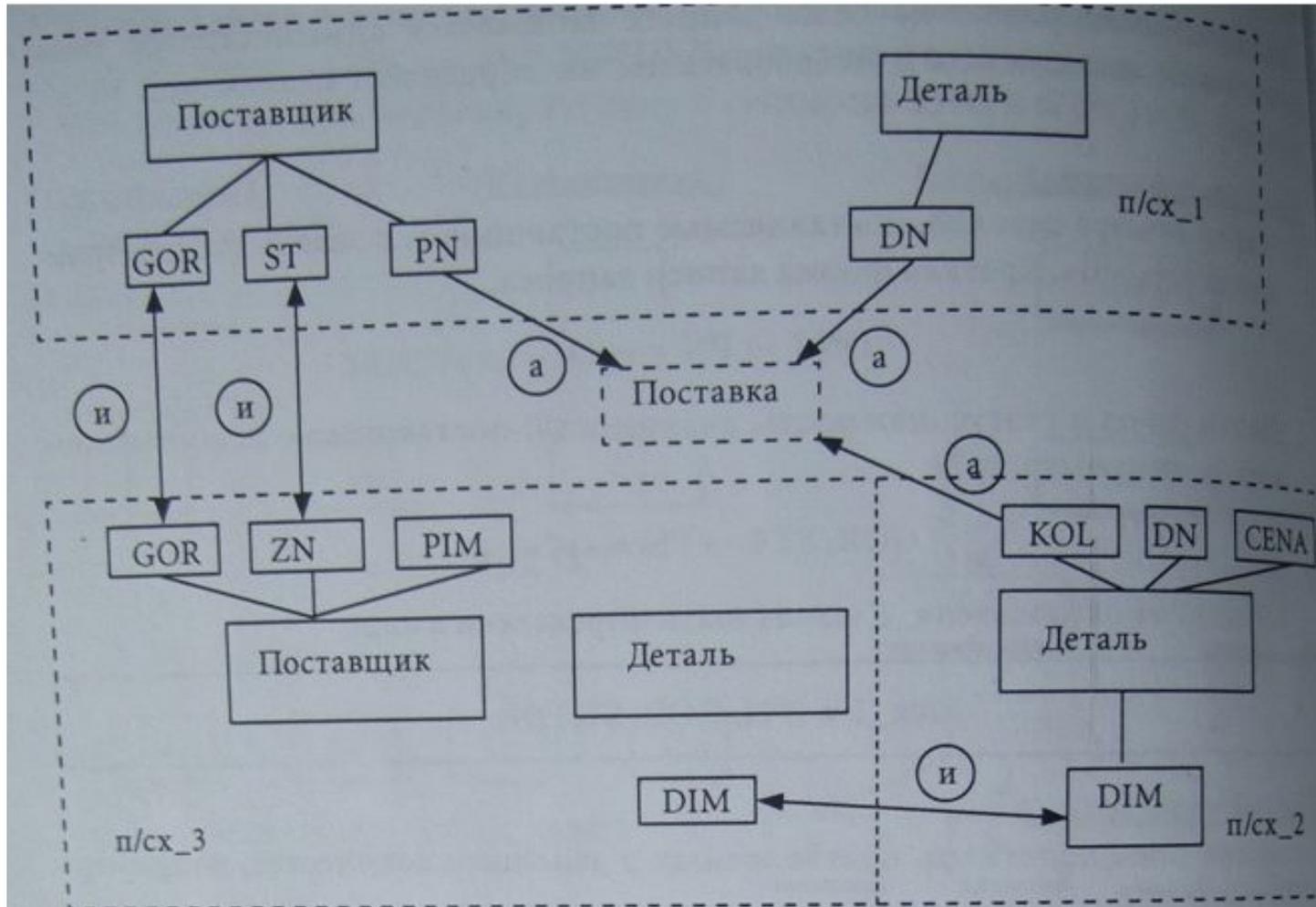
**Три концепции интеграции пользователей:** Идентичность, Агрегация(целое из частей), Обобщение(связано только с целыми).

Пример 2.1. имеем две сущности: ПОСТАВЩИК и ДЕТАЛЬ, между которыми существует связь М:М. Спроектируем базу данных для трех пользователей. Результат – СУБД независимое концептуальное представление данных.

Подсхемы пользователей:

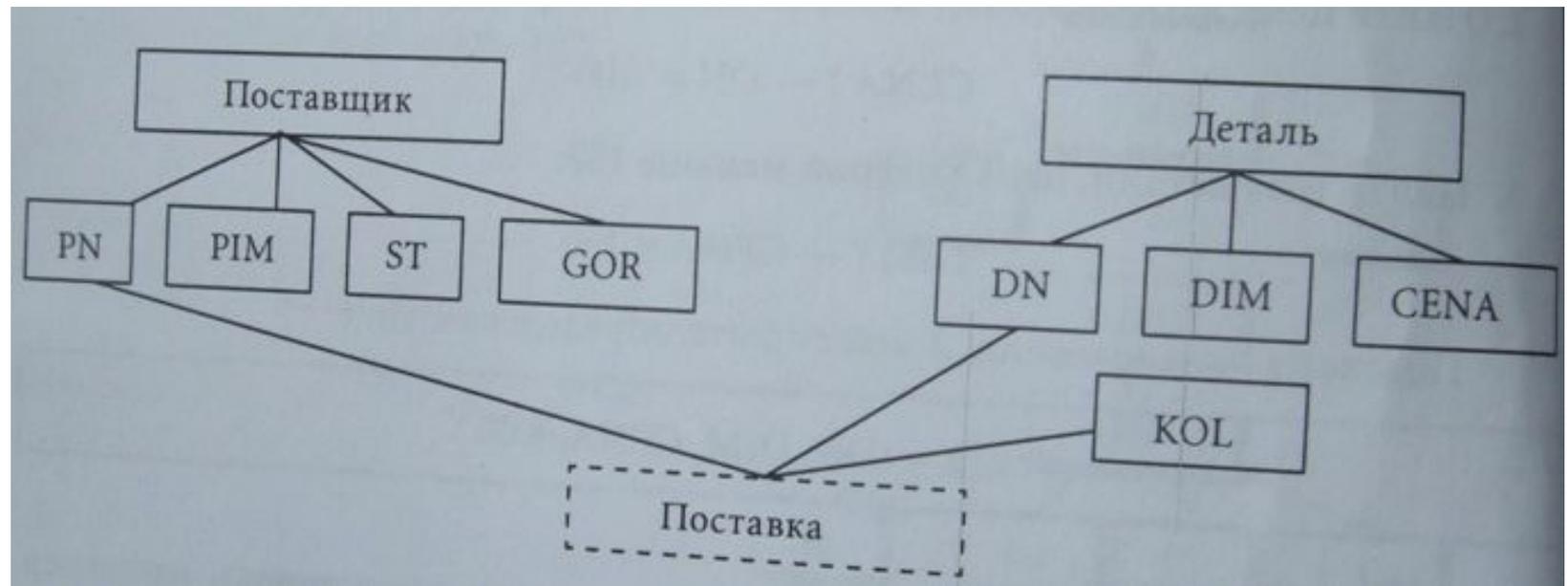
1. (PN,GOR,ST,DN)
2. (DN,DIM,CENA,KOL)
3. (PIM,ZN,GOR,DIM)

Пример 2.1. имеем две сущности: ПОСТАВЩИК и ДЕТАЛЬ, между которыми существует связь М:М. Спроектируем базу данных для трех пользователей. Результат – СУБД независимое концептуальное представление данных.



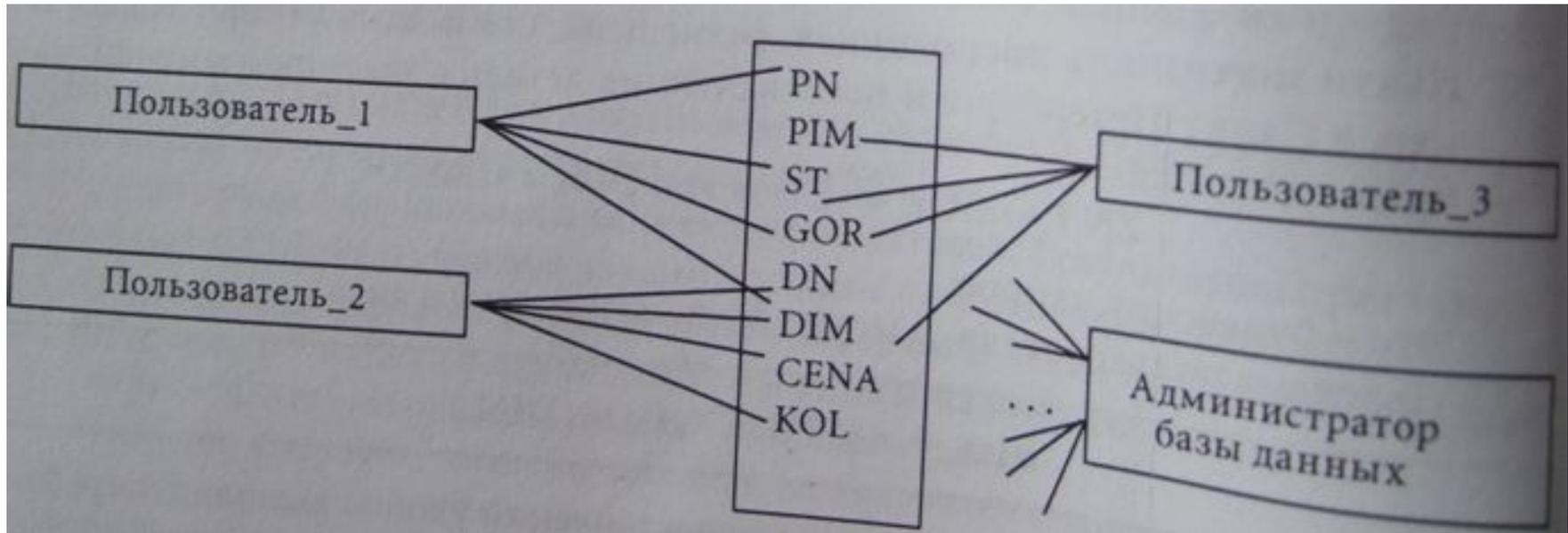
интеграция пользовательских представлений

Пример 2.1. имеем две сущности: ПОСТАВЩИК и ДЕТАЛЬ, между которыми существует связь М:М. Спроектируем базу данных для трех пользователей. Результат – СУБД независимое концептуальное представление данных.



Концептуальная схема

Пример 2.1. имеем две сущности: ПОСТАВЩИК и ДЕТАЛЬ, между которыми существует связь М:М. Спроектируем базу данных для трех пользователей. Результат – СУБД независимое концептуальное представление данных.



Права доступа пользователей к данным базы.

## 2.3 Проблемы проектирования СУБД-ориентированной реализации концептуального представления данных

Логический уровень:

- Сетевая;



- Иерархическая;



- Реляционная.

Отдел Начальник Сотрудник

Отдел Начальник Предприятие

Пример 2.2<...>Реализуем данные из этой предметной области в виде трех моделей.

Реляционная:

PPD( PN, PIM, ST, GOR, DN, DIM, CENA, KOL)							
p1	Ив.	80	МСК	d1	болт	80	100
p1	Ив.	80	МСК	d2	гайка	100	150
p2	Пет.	40	Самара	d1	болт	80	50
p2	Пет.	40	Самара	d2	гайка	100	100
p3	Сид.	100	СПб	d2	гайка	100	200

Post(PN, PIM, ST, GOR)				Det(DN,DIM,CENA)			PD(PN, DN, KOL)		
p1	Ив.	80	МСК	d1	болт	80	p1	d1	100
p2	Пет.	40	Самара	d2	гайка	100	p1	d2	150
p3	Кротов	100	СПб				p2	d1	50
							p2	d2	100
							p3	d2	200

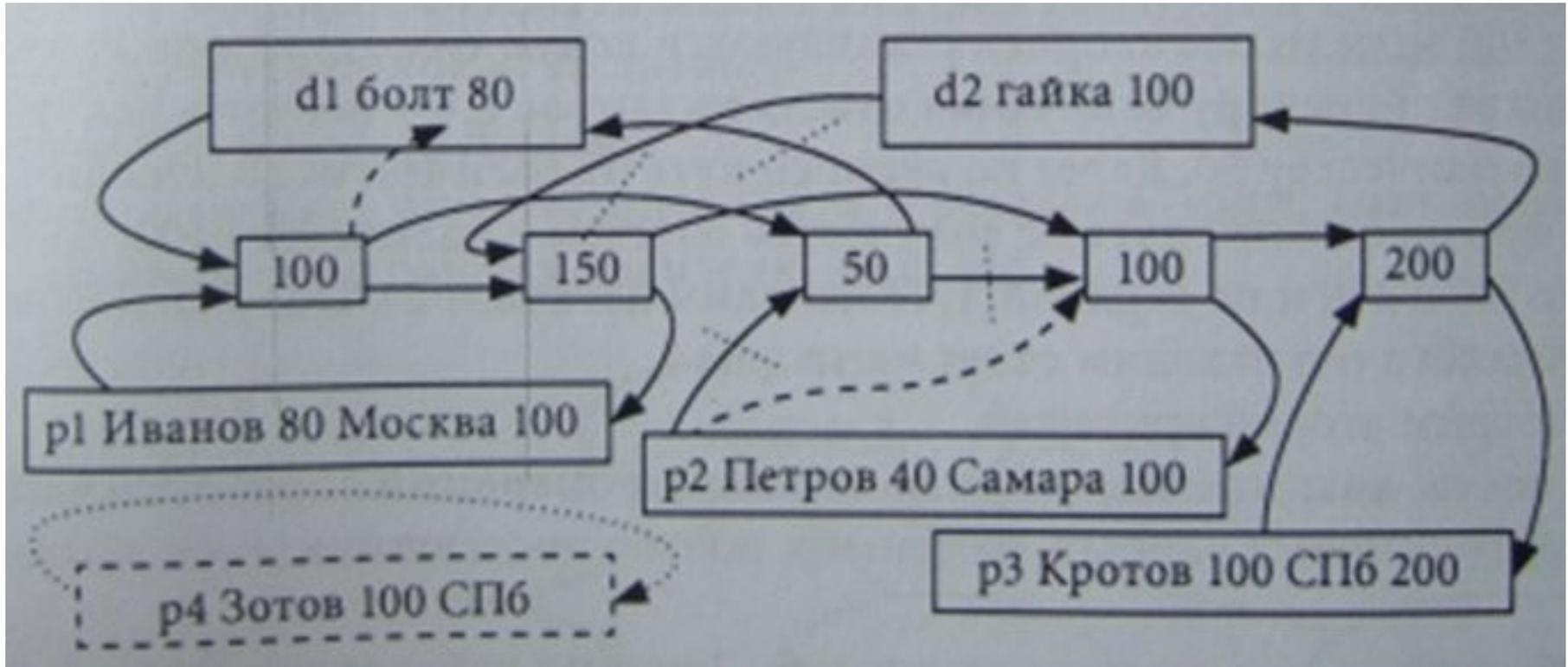
Пример 2.2<...>Реализуем данные из этой предметной области в виде трех моделей.

Иерархическая:

d1 болт 80					d2 гайка 100				
p1	Ив.	80	МСК	100	p1	Ив.	80	МСК	150
p2	Пет.	40	Самара	50	p2	Пет.	40	Самара	100
			p3	Кротов	100	СПб	200		

Пример 2.2<...>Реализуем данные из этой предметной области в виде трех моделей.

Сетевая:



# Сравнение моделей

Критерии:

способ реализации симметричных запросов

*DN? - PN="p2";*

*PN? - DN="d1"*

трудности, возникающие при совершении основных операции(добавление, удаление...)

*Поставщик("p4", "Зотов", 100, "СПб")*

# Часть 3. Теория проектирования реляционной реализации

# Определение отношения

Отношение	целое	строка		целое		Типы данных
	номер	имя	должность	деньги		Домены
	Табельный номер	Имя	Должность	Оклад	Премия	Атрибуты
	2934	Иванов	инженер	112	40	Кортежи
2935	Петров	вед. инженер	144	50		
2936	Сидоров	бухгалтер	92	35		

↑  
Ключ

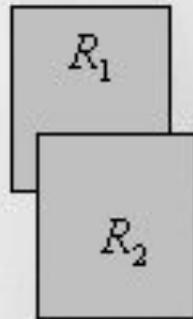
# Основные операции реляционной алгебры

- Объединение  $REZ = R \cup S$
- Вычитание  $REZ = R \setminus S$
- Декартово произведение  $REZ = R \times S$
- Проекция  $REZ = \pi_{i_1, \dots, i_m}(R)$
- Селекция  $REZ = \sigma_x(R)$
- Пересечение  $REZ = R \cap S$
- Деление  $REZ = R \div S$
- $\Theta$ -соединение  $REZ = R \bowtie_{L\theta M} S$
- Естественное  $REZ = R \bowtie S$

# Объединение

$$R \cup S = R \cup S$$

Пример:

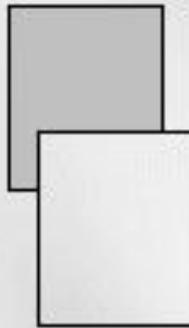


$$R[Q, T] \cup S = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \end{bmatrix} \cup \begin{bmatrix} \del{5} & \del{a} \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ \del{1} & \del{b} \end{bmatrix} = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \end{bmatrix}$$

# Вычитание

$$REZ = R \setminus S$$

Пример:

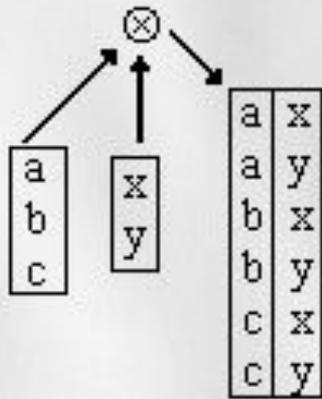


$$R[Q, T] - S = \begin{bmatrix} \del{5} & \del{a} \\ 3 & a \\ 9 & a \\ 1 & \del{b} \\ 2 & b \\ 4 & b \end{bmatrix} \cap \begin{bmatrix} \circled{5} & \circled{a} \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ \circled{1} & \circled{b} \end{bmatrix} = \begin{bmatrix} 3 & a \\ 9 & a \\ 2 & b \\ 4 & b \end{bmatrix}$$

# Декартово произведение

$$REZ = R \times S$$

Пример:

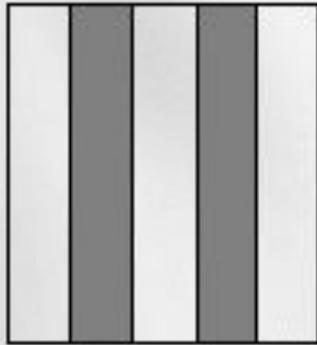


$$R[M, T] \otimes (R[Q, T] \cap S) = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \end{bmatrix} \otimes \begin{bmatrix} 5 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} x & a & 5 & a \\ x & a & 1 & b \\ y & a & 5 & a \\ y & a & 1 & b \\ z & a & 5 & a \\ z & a & 1 & b \\ w & b & 5 & a \\ w & b & 1 & b \end{bmatrix}$$

# Проекция

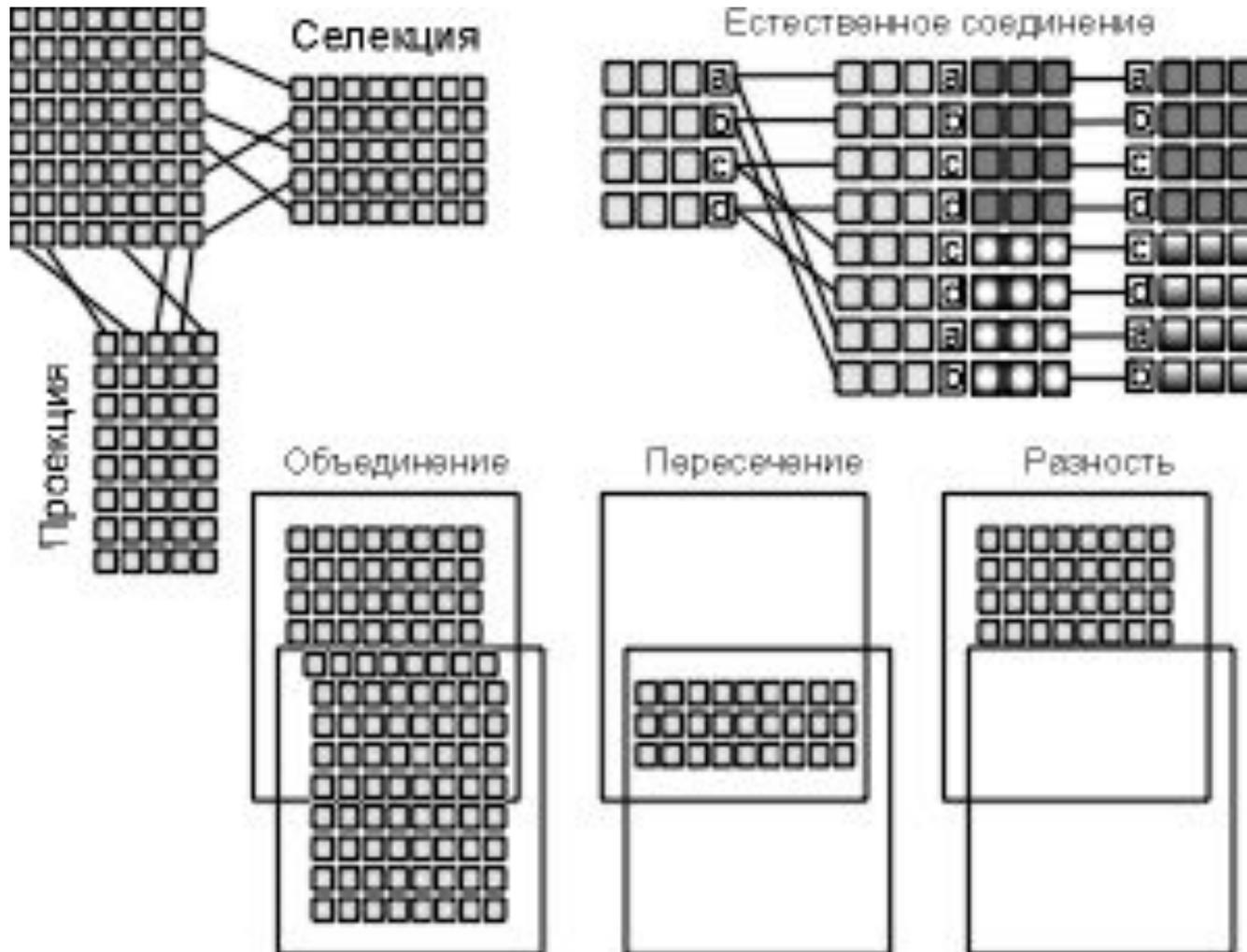
$$REZ = \pi_{i_1, \dots, i_m}(R)$$

Пример:



$$R[M, T] = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \\ \del w & \del b \\ \del w & \del b \end{bmatrix} = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \end{bmatrix}$$

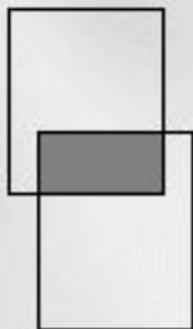
# +селекция



# Пересечение

$$REZ = R \cap S$$

Пример:



$$R[Q, T] \cap S = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \end{bmatrix} \cap \begin{bmatrix} 5 & a \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} 5 & a \\ 1 & b \end{bmatrix}$$

# Деление

$$REZ = R \div S$$

пусть

$$R_1(A_1, A_2) = \begin{bmatrix} a & x \\ a & y \\ a & z \\ b & x \\ c & y \end{bmatrix} \quad R_2(A_3, A_4) = \begin{bmatrix} 1 & x \\ 2 & x \\ 1 & y \end{bmatrix}$$

тогда

$$R_1[A_2 \div A_4] = \begin{bmatrix} a & x \\ a & y \\ a & z \\ b & x \\ c & y \end{bmatrix} \div \begin{bmatrix} x \\ y \end{bmatrix} = [a]$$

# ⊖-соединение

R = A B C    S = D E  
1 2 3        3 1  
4 5 6        6 2  
7 8 9

$REZ = R \langle_{B < D} \rangle S =$  A B C D E  
1 2 3 3 1  
1 2 3 6 2  
4 5 6 6 2

# Естественное соединение

Операция выполняется как эквисоединение отношений по общим атрибутам.

# Функциональные зависимости атрибутов

$R=XY$ , т.е.  $X \in R, Y \in R \quad X \rightarrow Y$

$r(R)=$

X	Y	Z
a	b	c
m	b	b
a	b	d

Полная функциональная зависимость

$r(R)=$  PN, DN, KOL, GOR

p1	d1	100	Москва
p2	d2	150	Москва
p2	d1	50	Самара
p2	d2	100	Самара
p3	d2	200	Москва

Пусть  $X=\{PN, DN\}$ ,  $Y=\{Kol\}$ .  $Y$  в полной ф/з от  $X$ ?

# Аксиомы и правила вывода функциональных зависимостей

- А1. Рефлексивности  $Y \subseteq X \subseteq Z$ , то  $X \rightarrow Y$   
 $X = \{A, B\}$ ,  $Y = \{A\}$ :  $AB \rightarrow A$
- А2. Пополнения  $X \rightarrow Y, Z \subseteq U$ , то  $XZ \rightarrow YZ$
- А3. Транзитивности  $X \rightarrow Y, Y \rightarrow Z$ , то  $X \rightarrow Z$
- А4. Коммутативности  $XY \rightarrow Z$ , то  $YX \rightarrow Z$
- А5. Свертка  $XX \rightarrow Z$ , то  $X \rightarrow Z$

Правило объединения:  $\forall \varnothing F; X, Y, Z \subseteq U; X \rightarrow Y, X \rightarrow Z \_ \_ \rightarrow$

Пр. псевдотранзитивности:  $\forall \varnothing Y, W \subseteq U; X \rightarrow Y, WY \rightarrow Z \_ \_ \rightarrow$

Правило декомпозиции.  
 $\forall \varnothing \rightarrow X, Z \subseteq Y \_ \_ \rightarrow$

**Пример.** Пусть схемой базы данных является график вылета самолетов из аэропорта, задаваемый в виде отношения со схемой ГРАФИК(ПИЛОТ, РЕЙС, ДАТА\_ВЫЛЕТА, ВРЕМЯ\_ВЫЛЕТА)

R(ГРАФИК)= ПИЛОТ, РЕЙС, ДАТА, ВРЕМЯ			
Иванов	83	10.08	10:15
Петров	281	10.08	15:30
Иванов	301	10.08	18:10
Петров	83	11.08	10:15

Поиск зависимостей перебором.

# Вычисление замыканий

Два вида замыканий:

Замыкание множества функциональных зависимостей на заданном множестве атрибутов.

$F^+$  Множество зависимостей  $X \rightarrow Y$ , каждая либо принадлежит  $F$ , либо следует из  $F$ , включая и тривиальные зависимости.

$F^+ = F$ -полное

Замыкание набора атрибутов  $X \subseteq U$  относительно множества функциональных зависимостей.

$X^+$  Множество атрибутов  $A$ , зависимости от  $X (X \rightarrow A)$  логически следуют из  $F$ .

# Алгоритм нахождения замыкания атрибутов $X^+$ относительно $F$ .

Исходные данные:

- Множество функ. зависимостей.
- Исходный набор атрибутов.

Суть: итерационно вычисление приближений:  $X^{(0)}, X^{(1)}, X^{(2)} \dots$

По следующим правилам:

1. начальное  $X^{(0)}$  задается как  $X$ ;
2. приближение  $X^{(j+1)}$  - как результат предыдущей итерации  $X^{(j)}$ , к которой добавлены атрибуты  $A_j \in Z$ , при которых в множестве  $F$  существуют зависимости вида  $X \rightarrow Z, X - Z \subseteq U^{(j)}, \subseteq$
3. Остановка  $X^{(s)} = X^{(s+1)}$

Пример. Пусть  $U=\{A,B,C,D,E,H,G\}$  и  $F=\{A-D,AB-DE,CE-G,E-H\}$ . Вычислить замыкание  $X=AB$

# Алгоритмы нахождения первичного ключа

- **Алгоритм исключения:**

**Пример.** Найдем первичный ключ отношения  $R=ABCDE$ , на котором определено множество функциональных зависимостей  $F=\{A-C, B-C, C-D, DE-C, CE-A\}$ .

**Алгоритм переборный.** Последовательный анализ претендентов на ключ – одиночных атрибутов, наборов из двух, трех...

При этом, если:

1) для какой-либо пары атрибутов  $A_i A_j$  верно:

$$A_i^+ \cup A_j^+ = U$$

то  $A_i A_j$  – первичный ключ схемы отношений.

2) для некоторого  $\exists X \in U \setminus Y_i \neq A_j$ ,  $Y_i = (X A_j)^+$   $A_i A_j$  – первичный ключ схемы отношений;

3) для некоторого  $\exists X \in U \setminus Y_{ij} \neq A_s$ ,  $Y_{ij} = (X A_j)^+$  то  $A_i A_j A_s$  – первичный ключ схемы отношений

4)....

# Эквивалентность множеств функциональных зависимостей атрибутов

Случаи, когда может использовать:

- 1) Разные разработчики проектировали, получали разные множества функциональных зависимостей.
- 2) Полученное множество по каким-либо причинам не устраивает разработчиков.

Множества эквиваленты, если равно их замыкания.

Проверка:

- каждую зависимость  $(X \rightarrow Y) \in F$  проверяют на принадлежность замыканию  $G^+$ .
- и наоборот.

Пример. Проверить эквивалентность:  $F = \{A-BC, A-D, CD-E\}$ ,  
 $G = \{A-BCE, A-BD, CD-E\}$

# Декомпозиция схем отношений

Декомпозицией схемы отношений  $R = A_1, \dots, A_k$  называется замена ее совокупностью  $\rho = \{R_1, R_2, \dots, R_n\}$  возможно пересекающихся подмножеств  $R_j \in \rho, j=1 \dots n$ , таких что  $R_1 \cup R_2 \cup \dots \cup R_n = R$ . Каждое множество  $R_j \subset R$  называется декомпозиционной подсхемой.

# Недостатки

PPD(	PN,	PIM,	ST,	GOR,	DN,	DIM,	CENA,	KOL)
p1	Ив.	80	МСК	d1	болт	80	100	
p1	Ив.	80	МСК	d2	гайка	100	150	
p2	Пет.	40	Самара	d1	болт	80	50	
p2	Пет.	40	Самара	d2	гайка	100	100	
p3	Сид.	100	СПб	d2	гайка	100	200	

Недостатки:

Потенциальная избыточность.

Потенциальная противоречивость.

Аномалии включения.

Аномалии удаления.

# Декомпозиция схем отношений

## Свойства:

Соединение без потерь информации относительно заданного множества функциональных зависимостей. *(Если отношение является естественным соединением его проекции на все декомпозиционные подсхемы).*

Сохранение исходного множества функциональных зависимостей на заданном множестве атрибутов.

**Пример.** Пусть R=ABC-схема отношений, на которой определено множество функциональных зависимостей F={A->B}. Проверим свойство соединения без потерь.

r={AB,AC} – обладает свойством соединения без потерь, покажем на произвольном выбранном экземпляре:

$r(R)=$	A B C	$\pi_{AB}(r)=$	<b>A B</b>	$\pi_{AC}(r)=$	<b>A C</b>
	a b c		a b		a c
	d b e		d b		d e

$\pi_{AB}(r) * \pi_{AC}(r) =$	$\pi_{AB}(r).A$	B	$\pi_{AC}(r).A$	C
	<b>a</b>	<b>b</b>	<b>a</b>	<b>c</b>
	a	b	d	e
	d	b	a	c
	<b>d</b>	<b>b</b>	<b>d</b>	<b>e</b>

$\pi_{AB}(r) \bowtie_A \pi_{AC}(r) =$	A B C	=	$r(R)$
	a b c		
	d b e		

# Алгоритмы проверки соединения без потерь информации

**Теорема.** Если  $\gamma=\{R_1, R_2\}$ -декомпозиция схемы отношения  $R$  на две подсхемы,  $F$ -мно-во функциональных зависимостей, определенных на  $R$ , то декомпозиция обладает свойством соединения без потерь информации относительно  $F$ , если и только если справедлива хотя бы одна из зависимостей:

$$(R_1 \cap R_2) \rightarrow (R_1 \setminus R_2) \in F^+ \quad \text{или} \quad (R_2 \cap R_1) \rightarrow (R_2 \setminus R_1) \in F^+$$

**Пример.** Пусть  $R=ABC$  и  $F=\{A \rightarrow B\}$

Обладают ли декомпозиции  $\gamma=\{AB, AC\}$ ,  $\gamma=\{AB, BC\}$  свойством соединения без потерь

**Теорема для декомпозиции, состоящей более чем из двух декомпозиционных подсхем...**

**Пример.** Пусть  $R=ABCDEFPS$  и  $F=\{B \rightarrow C, D \rightarrow EF, B \rightarrow PS, A \rightarrow CDPS, AP \rightarrow S\}$ . Выяснить обладает ли декомпозиция  $\gamma=\{AB, ACDPS, BCPS, DEF\}$  схемы отношения  $R$  свойством соединения без потерь информации?

**Пример.** Пусть  $R=ABCDEKM$  и  $F=\{A \rightarrow BC, A \rightarrow D, D \rightarrow EK, AD \rightarrow M, M \rightarrow AK\}$ . Выяснить обладает ли декомпозиция  $\gamma=\{ADME, EKC, ACE, DEB\}$  схемы отношения  $R$  свойством соединения без потерь информации?

# Свойство сохранения зависимостей.

Проекцией множества функциональных зависимостей  $F$  на множество атрибутов  $Z \subseteq U$ , обозначается как  $\pi_Z(F)$ , называется такое множество зависимостей  $(X \rightarrow Y) \in F^+$ , для которых  $XY \subseteq Z$ .

Пример.  $U=\{A,B,C\}$  и  $F=\{A \rightarrow B, B \rightarrow C\}$  и проекции  $\pi_{AB}(F)=\{A \rightarrow B\}$ ,  $\pi_{AC}(F)=\{A \rightarrow C\}$ .

Декомпозиция схемы отношений  $R$  сохраняет множество функциональных зависимостей  $F$ , определенное на  $R$ , если из объединения всех зависимостей, принадлежащих проекциям  $F$  на декомпозиционные подсхемы  $R_i$ ,  $i=1,2,\dots,k$ , логически следует все зависимости из  $F$ :

$$\left( \bigcup_{i=1}^k \pi_{R_i}(F) \right)^+ = F^+$$

Пример.  $U=\{A,B,C\}$  и  $F=\{A \rightarrow B, B \rightarrow C\}$ . Декомпозиция  $r=\{AB, BC\}$  – сохраняет фз,  $r=\{AB, AC\}$  – не сохраняет.

# Свойство сохранения зависимостей.

Алгоритмы:

- на основе алгоритма синтеза множества зависимостей;
- на основе расширенного множества зависимостей.

**Пример.** Пусть  $R=ABCDEM$  и  $F=\{A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow M, CE \rightarrow D\}$ . Декомпозиция  $r=\{R_1, R_2\}$ , где  $R_1=AEM$ ,  $R_2=ABCDE$ . Найдём проекции. Восстанавливаются ли зависимости?

**Пример.** Пусть  $R=ABCDEKX$  и  $F=\{A \rightarrow BC, A \rightarrow D, C \rightarrow B, D \rightarrow EK, AD \rightarrow X, X \rightarrow AK, E \rightarrow K, A \rightarrow K\}$ . Декомпозиции  $r=\{ADEX, EKC, ACEK, DEBC\}$ . Восстанавливаются ли зависимости?

Нормальные формы.

# Первая нормальная форма

Отношение со схемой  $F$  и множеством функциональных зависимостей  $F$  находится в 1НФ, если любой экземпляр схемы  $R$  удовлетворяет следующим условиям:

- каждый атрибут имеет уникальное имя,
- элементы кортежа с одним и тем же именем определены на одном и том же домене,
- элементы должны быть атомарны;
- каждый элемент кортежа должен иметь одно значение;
- в отношении не должно быть повторяющихся кортежей.

## Ненормализованное

PPD( PN, ST, GOR DN KOL )
p1 80 MCK d1 100
- - - d2 100
p2 40 Самара d1 300
- - - d2 40
p3 80 MCK d2 200
p3 100 СпБ d2 150

## Нормализованное

PPD( PN, ST, GOR DN KOL )
p1 80 MCK d1 100
p1 80 MCK d2 100
p2 40 Самара d1 300
p1 80 MCK d2 40
p3 80 MCK d2 200
p3 100 СпБ d2 150

## Недостатки 1НФ:

- Информационная избыточность
- Аномалии обработки данных(добавление, удаление, корректировка)

# Вторая нормальная форма

1НФ + каждый неключевой атрибут функционально полно зависит от любого возможного первичного ключа схемы отношений R.

$r(R_{11}) =$  PN ST GOR

p1 80 MCK

p2 40 Самара

p3 80 MCK

p3 80 MCK

$r(R_2) =$  PN DN KOL

p1 d1 100

p1 d2 100

p2 d1 300

p2 d1 40

p3 d2 200

p4 d2 150

$F_1 = \{PN \rightarrow GOR, GOR \rightarrow ST, PN \rightarrow ST\}$

$F_2 = \{(PN, DN) \rightarrow KOL\}$

# Третья нормальная форма

2НФ + каждый неключевой атрибут прямо, а не транзитивно зависит от любого функционально полно зависит от любого возможного первичного ключа схемы отношений R, а также все неключевые атрибуты отношения взаимно независимы (если они есть вообще).

$r(R_{11}) =$  PN    GOR  
p1    MCK  
p2    Самара  
p3    MCK  
p4    Спб

$r(R_{12}) =$  GOR ST  
MCK 80  
Самара 40  
СПб 100

# Нормальная форма Бойса-Кодда(НФБК)

Отношение  $R$  с множеством зависимостей  $F$  находится в НФБК, если левая часть каждой зависимости  $(X \rightarrow A)$  из  $F$ , где  $A$  не из  $F$ , есть возможный (потенциальный, альтернативный) первичный ключ.

# НФ более высоких порядков

**НФ4.** + для каждой многозначной зависимости, либо МЗ тривиальна, либо суперключ.

**НФ5.** + любая зависимость по соединению в нем определяется только его возможными ключами

# **Методы проектирования реляционной реализации концептуальной схемы**

# Методы проектирования

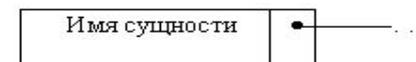
- Метод декомпозиции
- Метод синтеза
- Метод ER-диаграмм
- Комбинация методов

# Метод ER-диаграмм. Условные обозначения.

При проектировании методом ER - диаграмм обычно использует два вида диаграмм:

- диаграммы ER – экземпляров;
- диаграммы ER – типа, или просто ER – диаграммы.

Поставщик	Поставляет	Деталь
P1	—	D1
P2	—	D2
P3	—	D3
P4	—	D4
P5	—	D5
P6		D6



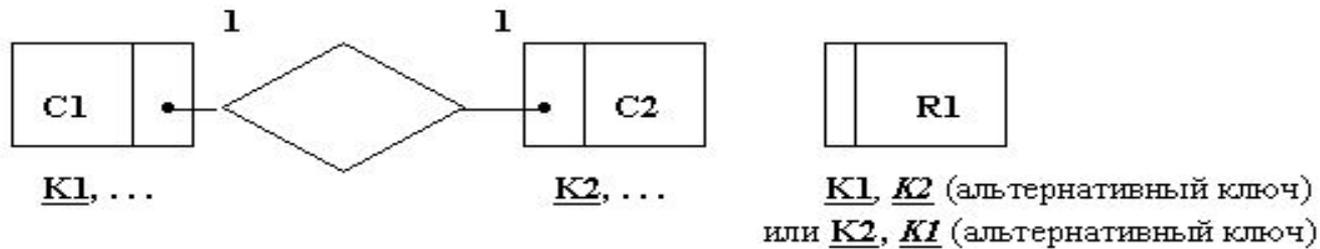
# Метод ER-диаграмм.

## Этапы проектирования

1. Выбор предметной области, выделение в ней сущностей с указанием возможных первичных ключей и связей между сущностями. Иногда сначала задают атрибуты сущностей, а затем на их основе определяют сами сущности.
2. Построение диаграмм ER-типа
3. Формирование набора предварительных отношений с указанием первичного ключа для каждого отношения. Это делается на основе восьми правил.
4. Добавление неключевых атрибутов в отношения.
5. Анализ предварительных отношений на предмет “сильных” нормальных форм. Если отношения не находятся в “сильной” нормальной форме, то проведение дальнейшей нормализации и получение окончательного набора отношений, находящихся в “сильной” нормальной форме.
6. Проверка выполнимости свойства соединения без потерь информации для окончательного набора отношений.

# Правило1.

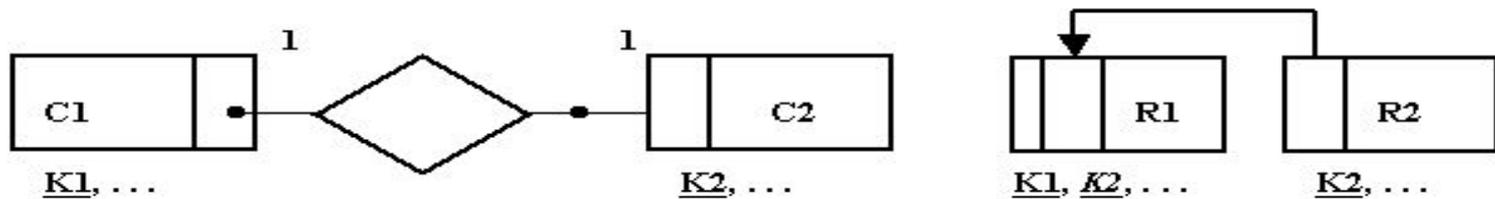
тип связи между парой сущностей 1 : 1 и КП: O – O



PN	PIM	DN	DIM
P1	Иванов	D1	гайка
P2	Петров	D2	шайба
P3	Сидоров	D3	шуруп
P4	Коновалов	D4	уголок

## Правило2.

тип связи между парой сущностей 1 : 1 и КП: О – Н или Н – О.



PN	PIM	DN	DIM
P1	Иванов	D1	гайка
P2	Петров	D2	шайба
P3	Сидоров	D3	шуруп
P4	Коновалов	D4	уголок
		D5	болт
		D6	подшипник

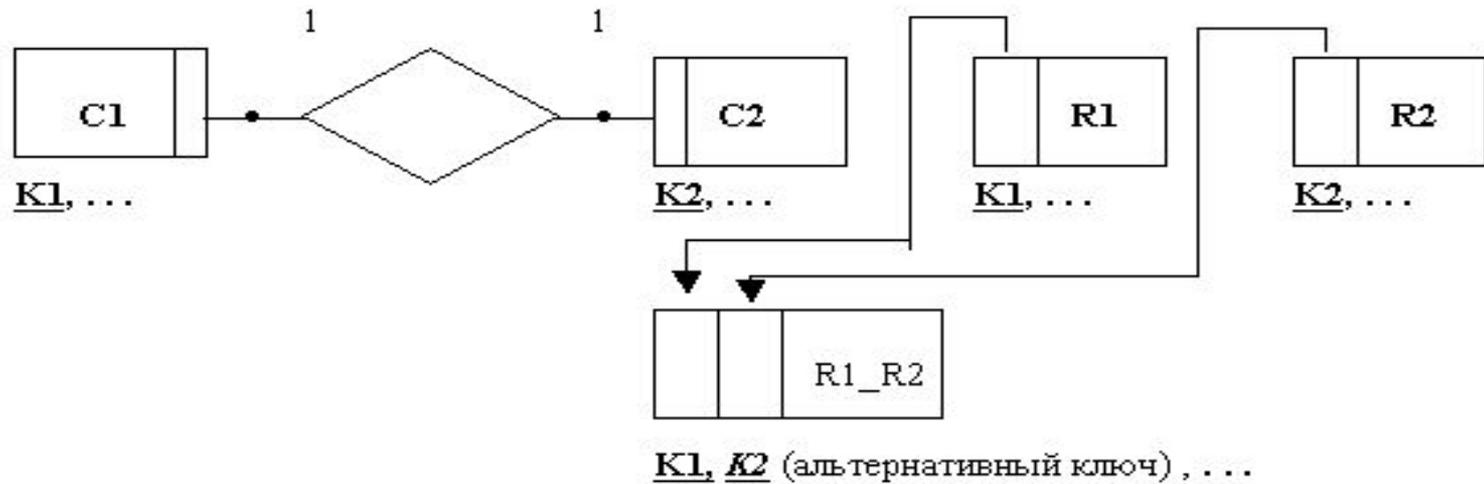
PN	PIM	DN
P1	Иванов	D1
P2	Петров	D2
P3	Сидоров	D3
P4	Коновалов	D4

DN	DIM
D1	гайка
D2	шайба
D3	шуруп
D4	уголок
D5	болт
D6	подшипник

# Правило 3.

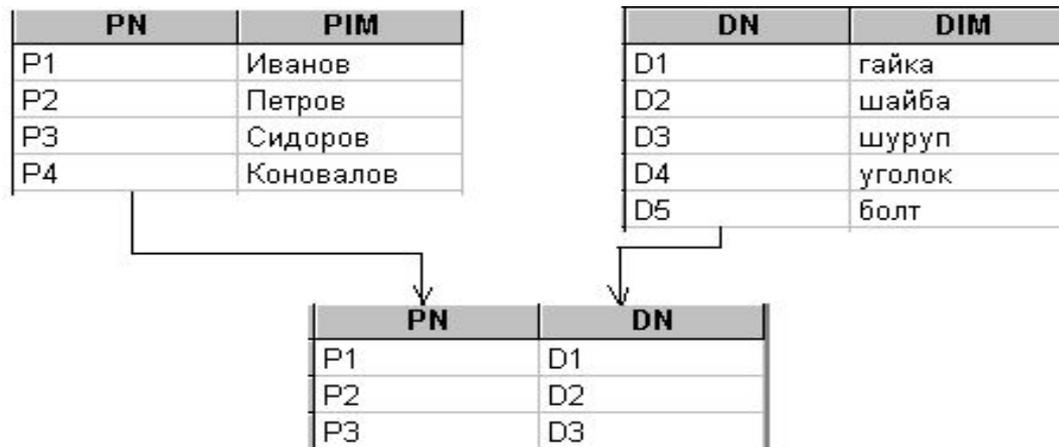
тип связи между парой сущностей 1 : 1 и КП: Н – Н



# Правило3.

тип связи между парой сущностей 1 : 1 и КП: Н – Н

PN	PIM	DN	DIM
P1	Иванов	D1	гайка
P2	Петров	D2	шайба
P3	Сидоров	D3	шуруп
		D4	уголок
		D5	болт
P4	Коновалов		



## Правило 4.

Тип связи между парой сущностей 1 : M или M : 1  
и тип связи - O для M – сущности

- формируются два отношения (по одному на каждую сущность) со своими первичными ключами. А далее первичный ключ 1-связной сущности добавляется как простой атрибут в отношение M-связной сущности.

Пусть **C1** – M-сущность с первичным ключом **K1** и класс принадлежности которой (КП) – обязательный (**O**), а **C2** – 1-связная сущность с ключом **K2**, КП которой может быть любым. Тогда согласно правилу 4 будут сформированы следующие отношения:

- R1(**K1**, K2, . . .)
- R2(**K2**, . . .)

# Правило 5.

Тип связи между парой сущностей 1 : М или М : 1

тип связи: Н для М – сущности

Например, пусть **C1** – 1-связная сущность с первичным ключом **K1**, класс принадлежности которой (КП) может быть любым, а **C2** – М-сущность с ключом **K2**, КП которой – необязательный (Н). Тогда:

R1(**K1**, ...)

R2(**K2**, ...)

R3(**K2**, K1, ...)

PN	PIM	DN	DIM
P1	Иванов	D1	гайка
P2	Петров	D2	шайба
P3	Сидоров	D3	шуруп
P1	Иванов	D4	болт
P1	Иванов	D5	уголок
P2	Петров	D6	подшипник
		D7	кронштейн

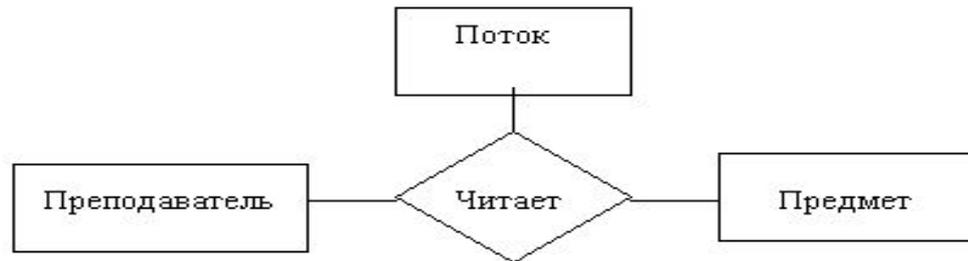


## Правило 6.

### Тип связи между парой сущностей M : M

- Всегда формируются три отношения. Два отношения будут соответствовать связываемым сущностям со своими первичными ключами, а третье – для связи. Первичный ключ отношения-связи будет составлен из первичных ключей обеих сущностей (составной первичный ключ).

## М-сторонняя связи



R1 (Номер преподавателя, . . .)

R2 (Название предмета, . . .)

R3 (Номер потока , . . .)

R4 (Номер преподавателя, Название предмета, Номер потока ,  
. . .)



# Правило 7

- В случае M-сторонней связи необходимо сгенерировать  $M + 1$  предварительное отношение – по одному отношению на каждую из  $M$  сущностей с ключами, совпадающими с ключами сущностей, и одно отношение для связи с составным ключом, составленным из первичных ключей связываемых сущностей. КП сущностей не имеют значения.

R1 (Номер преподавателя, . . .)

R2 (Название предмета, . . .)

R3 (Номер потока , . . .)

R4 (Номер преподавателя, Название предмета, Номер потока , . . .)

=>R4 (Название предмета, Номер потока, Номер преподавателя, . . .)

# Правила 8.

- Исходная сущность, порождающая роли, является источником одного предварительного отношения, причем ключ этой сущности служит ключом отношения. Ролевые элементы порождают еще по одному отношению с теми же ключами, что и исходная сущность. Связи, соединяющие ролевые отношения, генерируют столько отношений, сколько требуется для данного вида связи по соответствующим правилам.



# Примеры

## Этап 1.

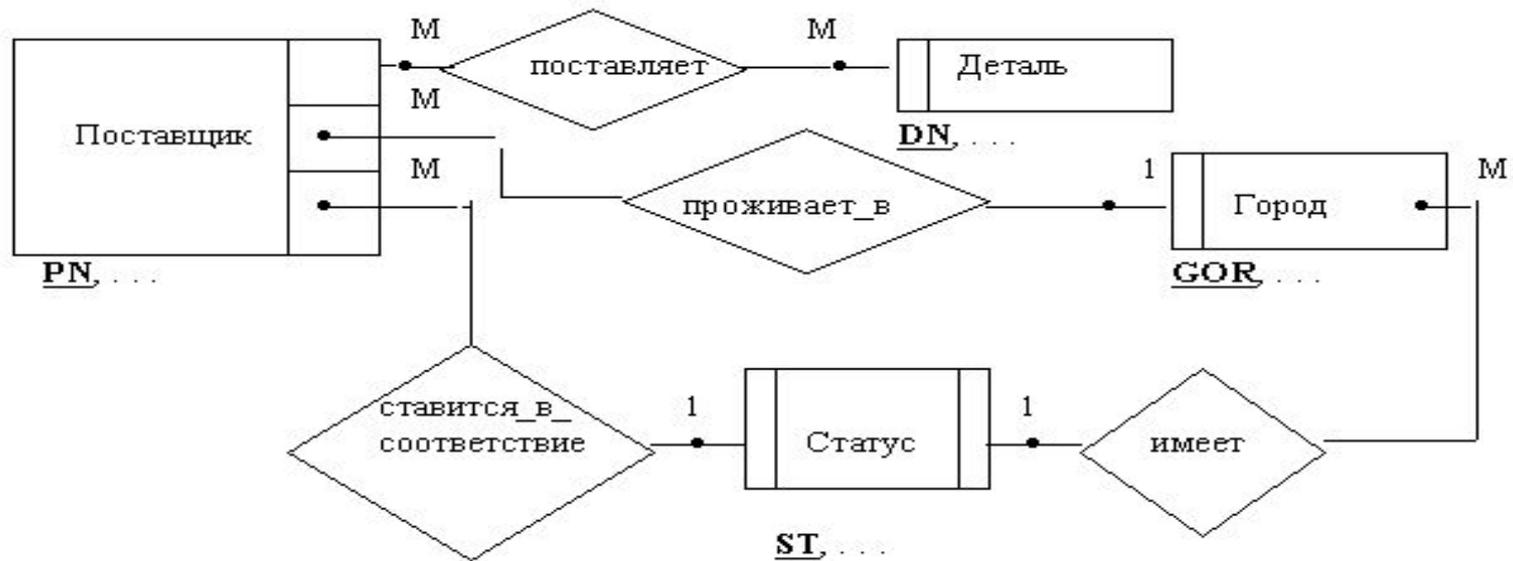
Выберем предметную область **ПОСТАВКА\_ДЕТАЛЕЙ**. Определим сущности и укажем первичный ключ для каждой сущности:

- **ПОСТАВЩИК** (PN, . . .)
- **ДЕТАЛЬ** (DN, . . .)
- **ГОРОД** (GOR, . . .)
- **СТАТУС** (ST, . . .)

Определим связи между парами сущностей:

- **ПОСТАВЩИК** *поставляет* **ДЕТАЛЬ**
- **ПОСТАВЩИК** *проживает\_в* **ГОРОДЕ**
- **ПОСТАВЩИКУ** *ставится\_в\_соответствие* **СТАТУС**
- **ГОРОД** *имеет* **СТАТУС**

## Этап 2.



- ПОСТАВЩИК (M, N)поставляет (M, N) ДЕТАЛЬ (правило 6)
- ПОСТАВЩИК (M, O)проживает\_в (1,Н) ГОРОДЕ (правило 4)
- ПОСТАВЩИКУ (M, O) ставится\_в\_соответствие (1, Н) СТАТУС (правило 4)
- ГОРОД (M, O) имеет (1, Н) СТАТУС (правило 4)

### Этап 3.

Для связи поставляет (правило 6):

- Поставщик (**PN**, . . .)
- Деталь (**DN**, . . .)
- Поставщик\_Деталь (**PN, DN**, . . .)

Для связи проживает\_в (правило 4):

- Поставщик (**PN**, **GOR**, . . .). Здесь **GOR** добавлен по правилу 4 как простой атрибут.
- Город (**GOR**, . . .)

Для связи имеет (правило 4):

- Город (**GOR**, **ST**, . . .). Здесь **ST** добавлен по правилу 4 как простой атрибут.
- Статус (**ST**, . . .)
- Для связи ставится\_в\_соответствие (правило 4):
- Поставщик (**PN**, **GOR**, **ST**, . . .) .  
Атрибут **GOR** уже был добавлен в связи проживает, а еще добавляется и **ST** как простой атрибут
- Статус (**ST**, . . .)

**Этап 4.** Добавляем в полученные предварительные отношения неключевые атрибуты.

- Поставщик (**PN**, PIM, GOR, ST), где PIM – имя поставщика
- Деталь (**DN**, DIM, CENA), где DIM – имя и CENA – цена детали
- Поставщик\_Деталь (**PN, DN**, KOL), где KOL – количество поставляемых деталей
- Город (**GOR**, ST)
- Статус (**ST**)

### Этап 5. Рассмотрим отношение **Поставщик**.

Для этого отношения справедливы следующие зависимости:

Поставщик = {PN → PIM, PN → GOR, PN → ST, GOR → ST}. Видно, зависимость PN → ST транзитивно следует из зависимостей PN → GOR и GOR → ST.

Тогда отношение **Поставщик** не в 3НФ. Что делать?

Декомпозиция на **Поставщик** (PN, PIM, GOR) и **Город** (GOR, ST). Проверь на сохранение функ. зависимостей и св-во соединением без потерь информации.

В итоге:

- Поставщик (PN, PIM, GOR)
- Деталь (DN, DIM, CENA)
- Поставляет (PN, DN, KOL)
- Город (GOR, ST)

В НФБК.

## Пример 5.

Этап 1. Определение предметной области и выделение сущностей и связей между ними.

Предметная область: **АВТОПРЕДПРИЯТИЕ.**

Сущности: **КЛИЕНТ** (Client), **ЗАКАЗ** (Zakaz), **ТЕХНИКА** (Car), **РАБОТНИКИ** (Worker), **ТИП** (Tip).

Атрибуты: код клиента (Code\_cl), название фирмы (Name\_cl), адрес фирмы (Adress\_cl), телефон фирмы (Teleph\_cl), факс фирмы (Fax\_cl), ИНН фирмы (INN\_cl), код работника (Code\_w), фамилия работника (F\_w), имя работника (N\_w), отчество работника (O\_w), год рождения работника (Year\_w), разряд работника (Raz\_w), стаж работника (St\_w), код техники (Code\_c), марка техники (Marka\_c), государственный номер машины (GN\_c), год выпуска (Year\_c), номер двигателя машины (Dwig\_c), цвет машины (Color\_c), код типа (Code\_tip), тип (Tip), код заказа (Code\_z), дата заказа (Date\_z), адрес доставки заказа (Adrd\_z).

Client (**Code\_cl**, ...)

Car (**Code\_c**, ...)

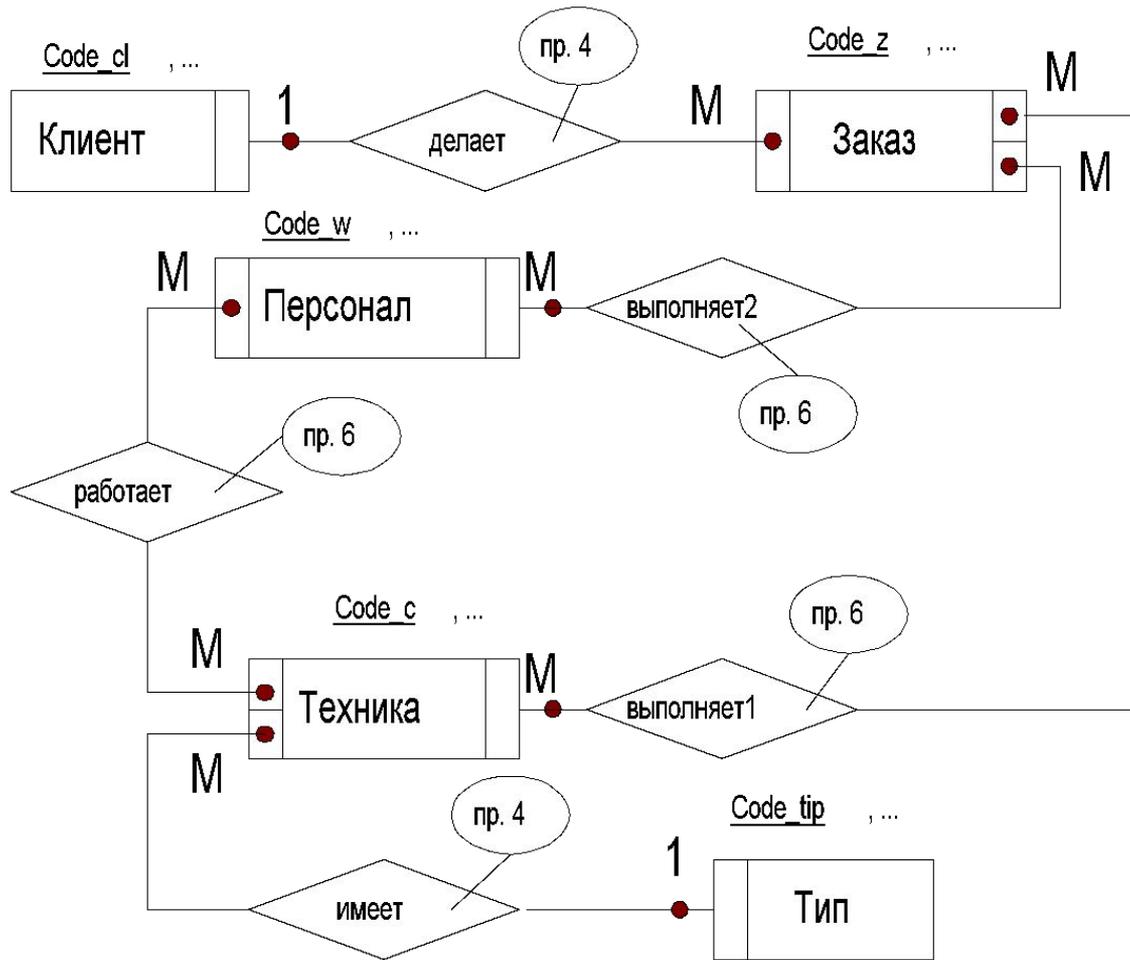
Worker (**Code\_w**, ...)

Tip (**Code\_tip**, ...)

Zakaz (**Code\_z**, ...)

Клиент	(1, Н)	<i>делает</i>	(М, О)	заказ (правило 4)
Техника	(М, О)	<i>имеет</i>	(1,О)	тип (правило 4)
Техника	(М, Н)	<i>выполняет1</i>	(М, О)	заказ (правило 6)
Персонал	(М, Н)	<i>выполняет2</i>	(М, О)	заказ (правило 6)
Персонал	(М, О)	<i>работает</i>	(М, О)	техника (правило 6)

**Этап 2.** Построение ER-диаграмм типа с учетом всех сущностей и связей между ними.



### Этап 3. Формирование набора предварительных отношений с указанием первичных ключей для них.

Для связи *делает* (правило 4) – два отношения:

Client (**Code\_cl**, ...)

Zakaz (**Code\_z**, Code\_cl, ...)

Для связи *имеет* (правило 4) – два отношения:

Tip (**Code\_tip**, ...)

Car (**Code\_c**, Code\_tip, ...)

Для связи *выполняет1* (правило 6) – три отношения:

Car (**Code\_c**, Code\_tip, ...)

Zakaz (**Code\_z**, Code\_cl, ...)

Zakaz\_Car (**Code\_z, Code\_c**, ...)

Для связи *выполняет2* (правило 6) – три отношения:

Worker (**Code\_w**, ...)

Zakaz (**Code\_z**, Code\_cl, ...)

Zakaz\_Worker (**Code\_z, Code\_w**, ...)

Для связи *работает* (правило 6) – три отношения:

Worker (**Code\_w**, ...)

Car (**Code\_c**, Code\_tip, ...)

Car\_Worker (**Code\_c, Code\_w**, ...)

## Этапы 4 и 5. Добавление неключевых отношений и анализ полученных отношений.

Client (Code\_cl, Name\_cl, Adress\_cl, Teleph\_cl, Fax\_cl, INN\_cl)

Worker (Code\_w, F\_w, N\_w, O\_w, Year\_w, Raz\_w, St\_w)

Car (Code\_c, Code\_tip, Marka\_c, GN\_c, Year\_c, Dwig\_c, Color\_c)

Tip (Code\_tip, Tip)

Car\_Worker (Code\_c, Code\_w)

Zakaz (Code\_z, Code\_cl, Date\_z, Adrd\_z)

Zakaz\_Car (Code\_z, Code\_c)

Zakaz\_Worker (Code\_z, Code\_w)