

МОДЕЛИРОВАНИЕ ДАННЫХ.

ДИАГРАММА "СУЩНОСТЬ-

СВЯЗЬ"

ЛЕКЦИЯ 2

Разработчик профессор,
Заслуженный работник науки и
образования Гребенюк И.И.

Для логического проектирования реляционных ХД применяются следующие методики.

Метод моделирования "сущность-связь" (ER modeling) дает абстрактную модель *предметной области*, используя следующие основные понятия: *сущности* (entities), *взаимосвязи* (relationships) между сущностями

Метод многомерного моделирования (*Dimensional modeling*) дает абстрактную модель *предметной области*, используя следующие основные понятия: *показатели* или *метрики* (measures), *факты* (facts) и *измерения* (dimensions).

Методы моделирования временных данных (*Temporal data modeling*) дают абстрактную модель *фрагмента предметной области*, представляющего временные ряды данных, и используют следующие основные понятия: *временные метки* (timestamps), *временной ряд* (time series), дата, диапазон дат, классы.

Метод моделирования "свод данных" (Data Vault) дает абстрактную модель *фрагмента предметной области*, основываясь на математических принципах нормализации *отношений*, и использует следующие основные понятия: *сущности-концентраторы* (Hub Entities), *связывающие сущности* (Link Entities), *сущности-спутники* (Satellite Entities).

ВОПРОС 1

ПОНЯТИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И АРХИТЕКТУРА ДАННЫХ

Классификация объектов предметной области



Имя

Ситуации

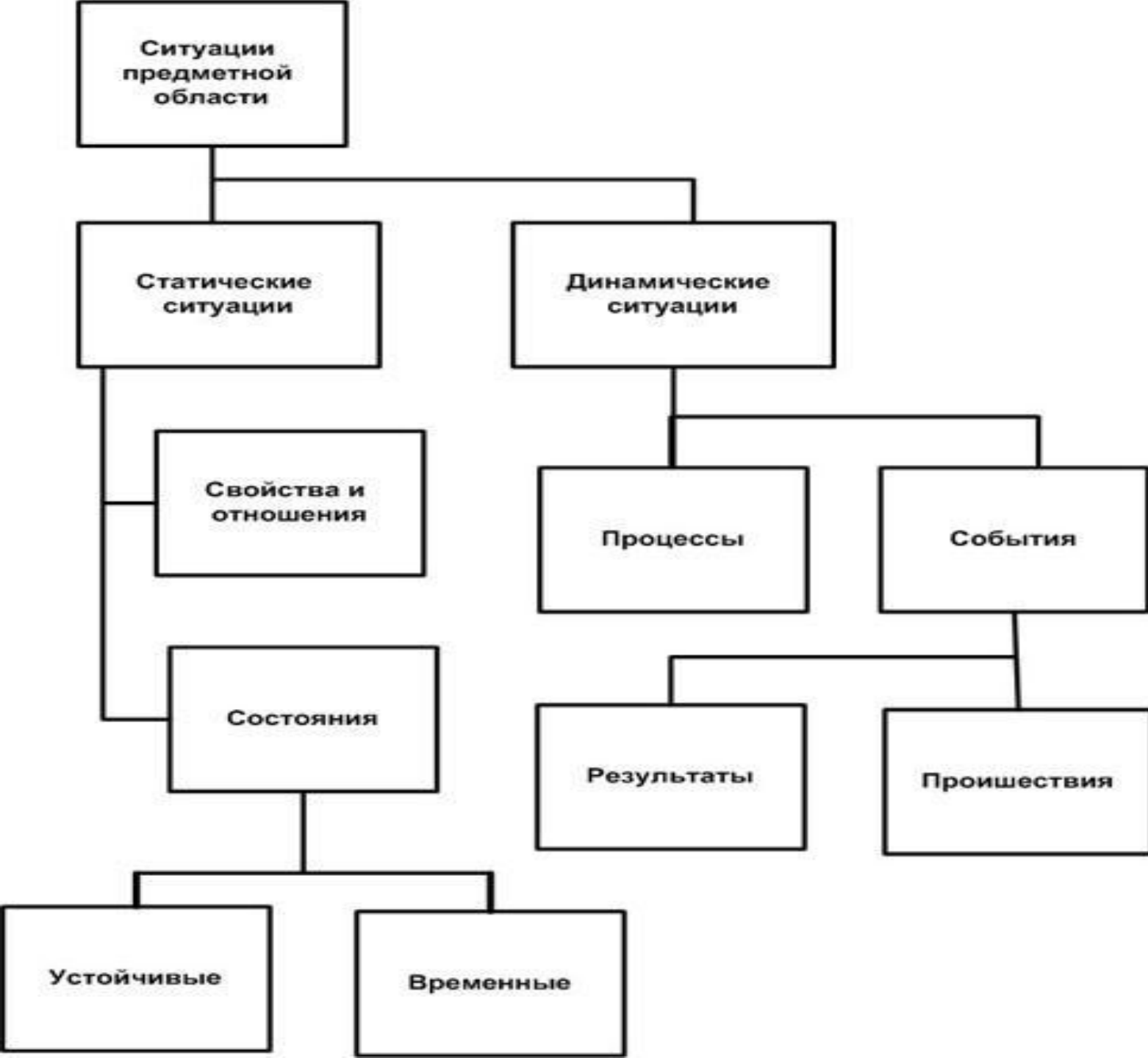
Методы
математической
ЛОГИКИ

• **этер**
жден
и ви
предс
тавит
как в
виде,
вразд
мощ
новше
он
впли
снос
вжж
л
иден
оо
тифе
ктам
капи

Рассмотрим высказывание: "Студент Иванов А.А, родился в 1997 году". Оно выражает следующие свойства объекта "Иванов А.А.":
в явном виде — год рождения;
в неявном виде – принадлежность к студентам.

РОДИЛСЯ (Иванов А.А., 1997)

**ЯВЛЯЕТСЯ СТУДЕНТОМ (Иванов А.
А.)**



Классификация ситуаций предметной области

Различают статические и динамические ситуации.

- Примерами статических ситуаций являются такие ситуации, как "иметь цвет", "иметь возраст" и т.д.
- Примерами динамических ситуаций являются такие ситуации, как "создать механизм", "выпечь

ВОПРОС 2

АРХИТЕКТУРА ДАННЫХ ПРЕДМЕТНОЙ ОБЛАСТИ

ОПРЕДЕЛЕНИЯ

Архитектура данных — это принципы субъективного представления информации в виде данных в рамках модели предметной области.

Одним из ключевых моментов построения архитектуры данных является *степень детализации информации* при преобразовании ее в элементы данных. Процесс такого преобразования называют **структуризацией данных**.

Уровень структуризации данных — это степень детализации хранимых данных, оптимальная с точки зрения решения информационно-аналитических задач в рамках предметной области ХД.

Одной из задач проектировщика ХД на уровне *логического моделирования* является принятие решения об оптимальном

Для ХД характерны три основных вида данных (класса).

Фактические

данные (Real-time data) представляют собой текущее состояние количественных и качественных показателей деятельности организации. Источником таких данных являются обычно OLTP-системы. Таким данным присущ высокий уровень структуризации. Для того чтобы использовать такие данные в ХД, их нужно предварительно обработать с помощью процедур очистки.

Производные данные (Derived data) представляют собой данные, которые получены в результате суммирования, агрегации и усреднения фактических данных. В зависимости от задач анализа такие данные могут быть либо детальными, либо итоговыми.

Консолидированные данные (Reconciled data) — это фактические данные, которые были очищены и представляют собой интегрированный источник данных для решения задач анализа. Основное требование к таким данным — их согласованность (consistency).

ВОПРОС 3

**ПОНЯТИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И
ХРАНИЛИЩА ДАННЫХ**

Понятие *предметной области* используется практически при проектировании и разработке всех классов информационных систем с базами и хранилищами данных.

Предметная область определяет ту часть реального мира, которая будет моделироваться и реализоваться в системе.

Предметная область определяет наиболее общие вытекающие из ее семантики критерии и требования к системе.

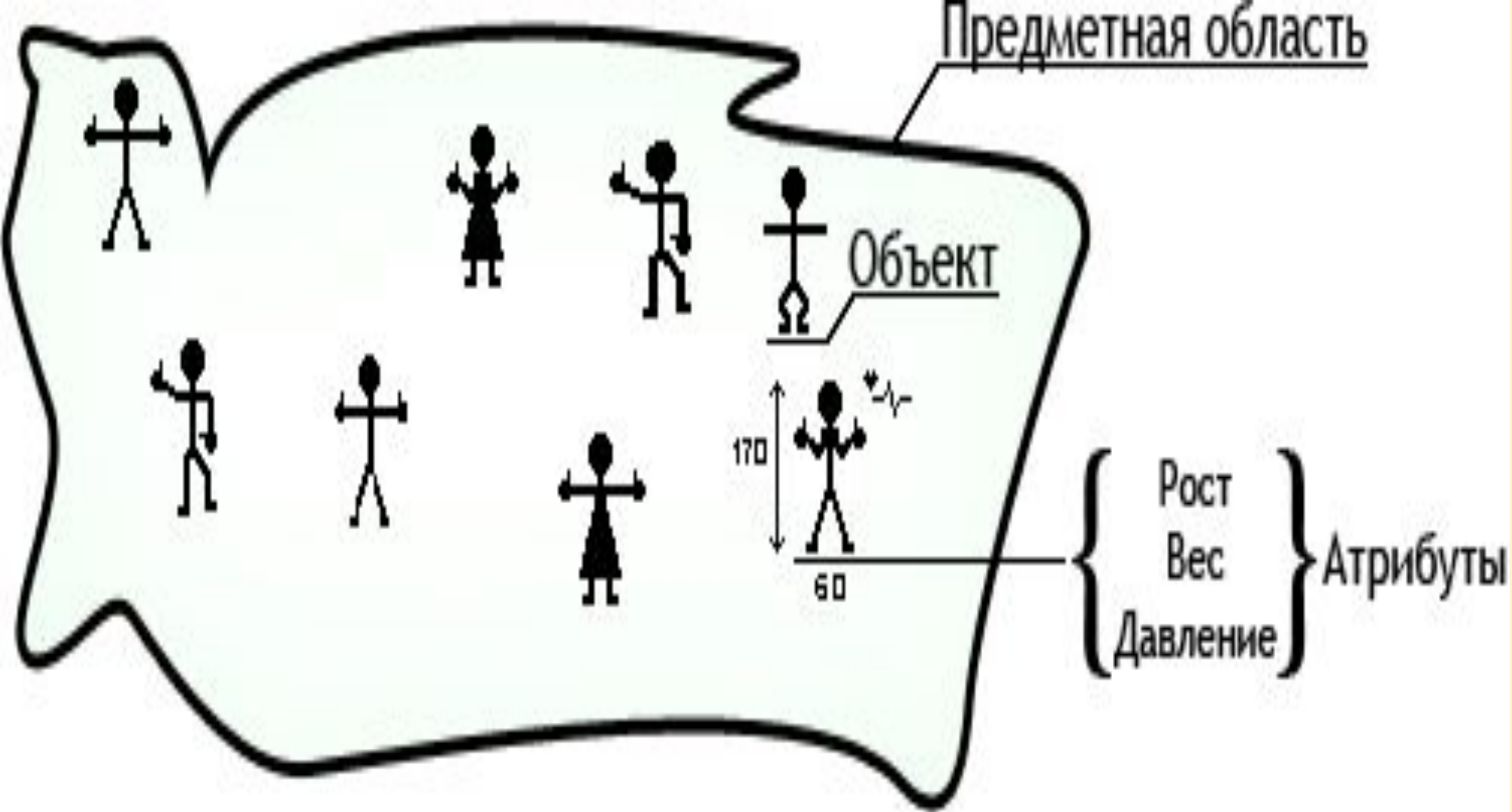
Предметная область

Объект

170
60

Рост
Вес
Давление

Атрибуты



- Для выделения *предметных областей* в ХД часто используется так называемая методика "правило SW1", а именно ответы на вопросы: когда (when), где (where), кто (who), что (what), почему (why) и как (how) – по отношению к видам деятельности организации (интересы бизнеса). Например, при ответе на вопрос "кто" интересы бизнеса могут охватывать следующие объекты: "покупатели", "сотрудники", "поставщики", "менеджеры", "партнеры по бизнесу" и т. д.

Вид VI– системы

Ретроспективные

Аналитические

Предсказательные

На чем основывается

Анализ данных о прошлых операциях

Объединяют и анализируют данные из внутренних и внешних источников

Итеративный процесс сбора информации из различных источников

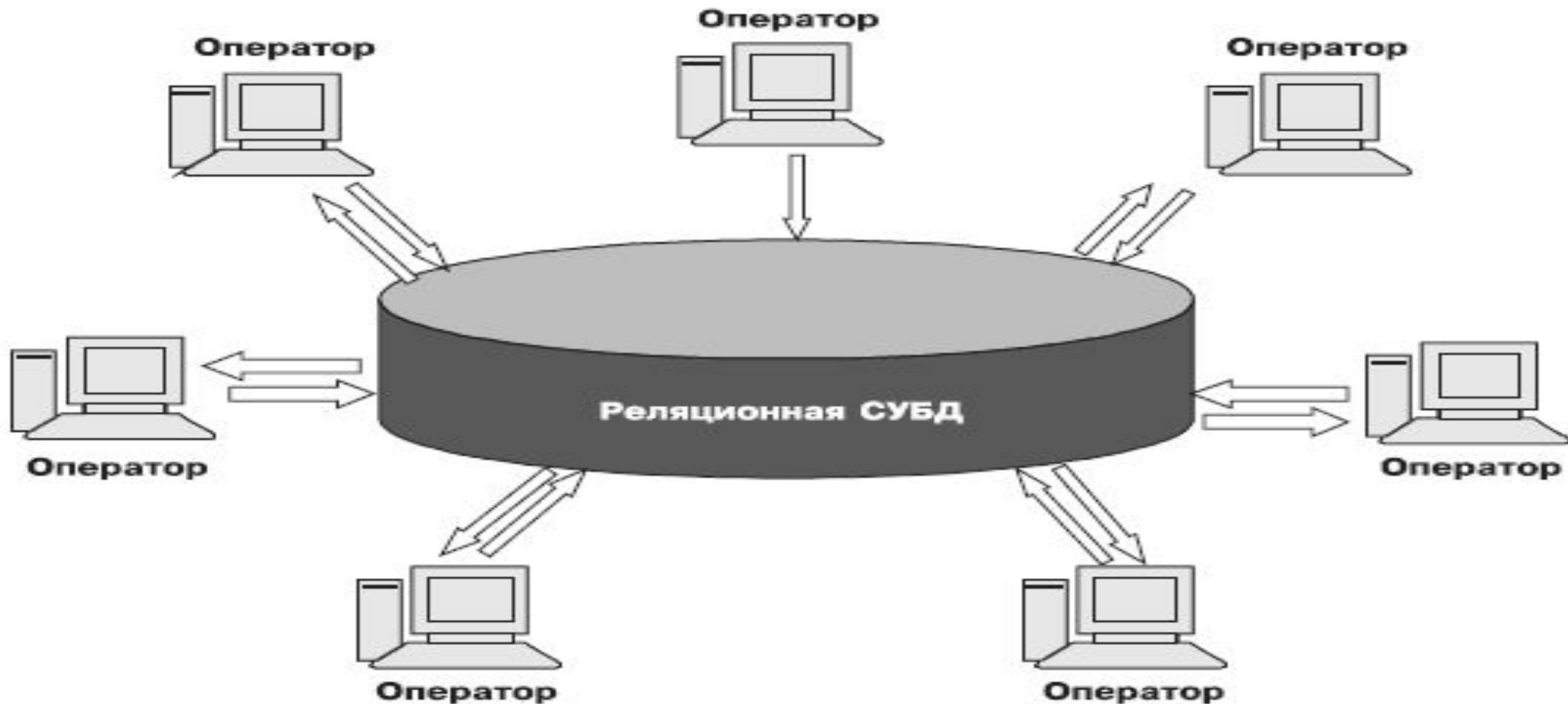
Назначение

Позволяет лицу, ответственному за принятие решения, видеть, какие действия в прошлом дали положительный результат, а какие — нет

Анализ данных о прошлых операциях

Анализ данных о прошлых операциях

При решении задач анализа и, следовательно, при разработке BI-систем наиболее перспективным подходом для определения *предметной области* является изучение бизнес-процессов организации, а не функции, как в случае OLTP-систем.



<i>Домен</i>	Совокупность допустимых значений
<i>Кортеж</i>	Таблица
<i>Кардинальность</i>	Количество строк в таблице
<i>Атрибут</i>	Поле, столбец таблицы
<i>Степень отношения</i>	Количество полей (столбцов)
<i>Первичный ключ</i>	Уникальный идентификатор

Внешний ключ, как и первичный ключ, тоже может представлять собой комбинацию столбцов. На практике внешний ключ всегда будет составным (состоящим из нескольких столбцов), если он ссылается на составной первичный ключ в другой таблице. Количество столбцов и их типы данных в первичном и внешнем ключах совпадают.

Внешний ключ — это столбец или подмножество одной таблицы, который может служить в качестве первичного ключа для другой *таблицы*. *Внешний ключ* таблицы является ссылкой на первичный ключ другой таблицы. Правило ссылочной целостности гласит, что внешний ключ может быть либо пустым, либо соответствовать значению первичного ключа, на который он ссылается. Внешние ключи являются неотъемлемой частью реляционной модели, поскольку реализуют связи между таблицами базы данных.

Модель предъявляет к таблицам следующие требования:

- 1) данные в ячейках таблицы должны быть структурно неделимыми
- 2) данные в одном столбце должны быть одного типа;
- 3) каждый столбец должен быть уникальным (недопустимо дублирование столбцов);
- 4) столбцы размещаются в произвольном порядке;
- 5) строки размещаются в таблице также в произвольном порядке;
- 6) столбцы имеют уникальные наименования.

Концепция реляционной модели определяется следующими двенадцатью правилами.

Правило единственности. Если в реляционной системе есть низкоуровневый язык (обрабатывающий одну запись за один раз), то должна отсутствовать возможность использования его для того, чтобы обойти правила и условия целостности, выраженные на реляционном языке высокого уровня (обрабатывающем несколько записей за один раз).

- Правило 2 указывает на роль первичных ключей при поиске информации в базе данных. Имя таблицы позволяет найти требуемую таблицу, имя столбца позволяет найти требуемый столбец, а первичный ключ позволяет найти строку, содержащую искомый элемент данных.
- Правило 3 требует, чтобы отсутствующие данные можно было представить с помощью недействительных значений (NULL).
- Правило 4 гласит, что реляционная база данных должна сама себя описывать. Другими словами, база данных должна содержать набор *системных таблиц*, описывающих структуру самой базы данных.
- Правило 5 требует, чтобы СУБД использовала язык реляционной базы данных, например SQL. Такой язык должен поддерживать все основные функции СУБД — создание базы данных, чтение и ввод данных, реализацию защиты базы данных и т. д.

- Правило 6 касается *представлений*, которые являются виртуальными таблицами, позволяющими показывать различным пользователям различные фрагменты структуры базы данных. Это одно из правил, которые сложнее всего реализовать на практике.
- Правило 7 акцентирует внимание на том, что базы данных по своей природе ориентированы на множества. Оно требует, чтобы операции добавления, удаления и обновления можно было выполнять над множествами строк. Это правило предназначено для того, чтобы запретить реализации, в которых поддерживаются только операции над одной строкой.
- Правила 8 и 9 означают отделение пользователя и прикладной программы от низкоуровневой реализации базы данных. Они утверждают, что конкретные способы реализации хранения или доступа, используемые в СУБД, и даже изменения структуры таблиц базы данных не должны влиять на возможность пользователя работать с данными.
- Правило 10 гласит, что язык базы данных должен поддерживать ограничительные условия, налагаемые на вводимые данные и действия, которые могут быть выполнены над данными.
- Правило 11 гласит, что язык базы данных должен обеспечивать возможность работы с распределенными данными, расположенными на других компьютерных системах.
- Правило 12 предотвращает использование других возможностей для работы с базой данных, помимо языка базы данных, поскольку это может нарушить ее целостность.

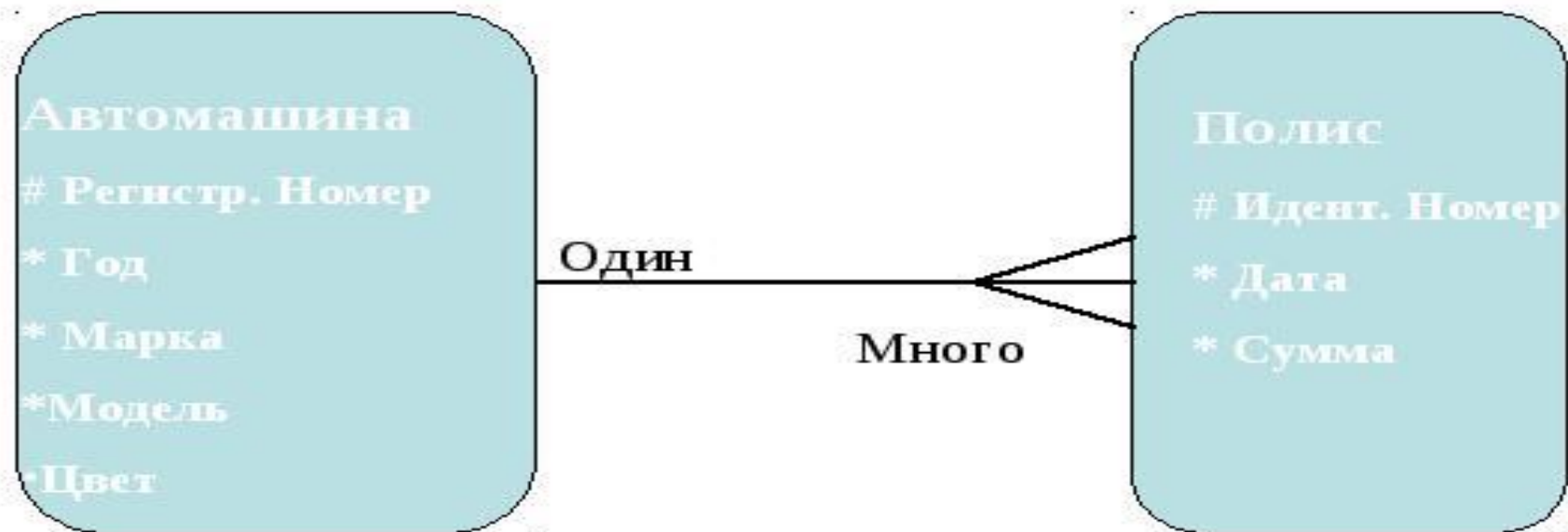
ВОПРОС 4

**МОДЕЛИРОВАНИЕ МЕТОДОМ
"СУЩНОСТЬ-СВЯЗЬ"**

Диаграммы ERD - «сущность-связь»

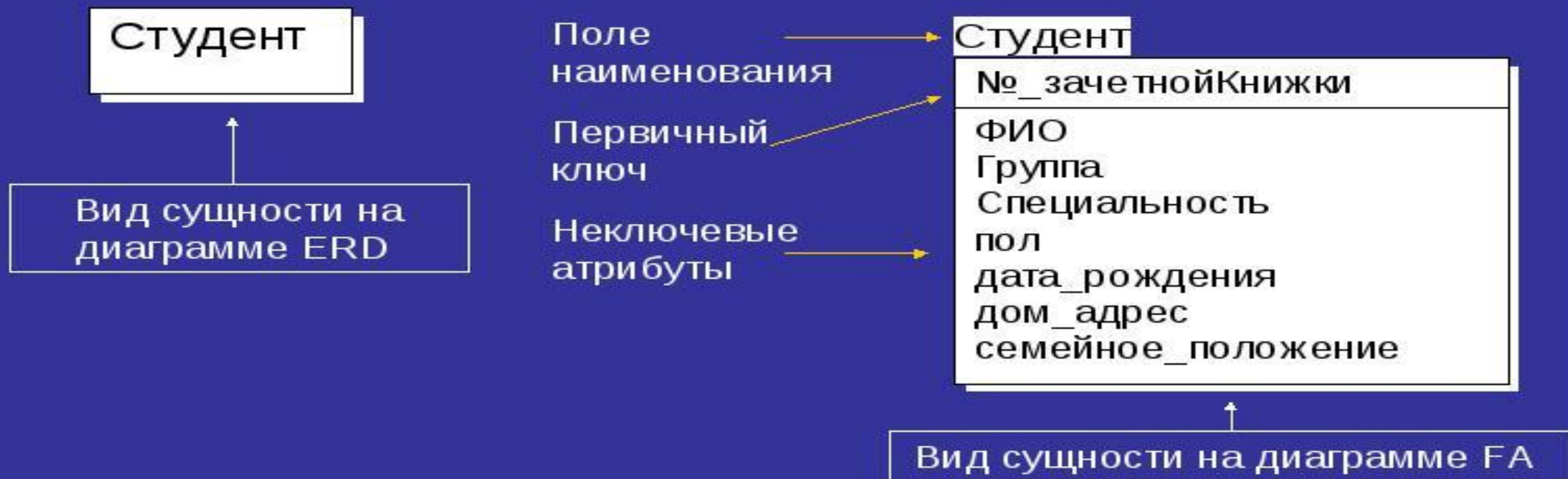
Описывают структуры данных, связанных с различными объектами модели; документируют сущности процесса (их идентификаторы, атрибуты) и способы взаимодействия между ними.

(Нотация Баркера)



Графическое представление сущности

Различают следующие уровни представления сущности: **диаграмма «сущность-связь» (ERD)**, **модель данных, основанная на ключах (KB)**, **полная атрибутивная модель (FA)**



Для проектировщика важно знать, что все экземпляры сущности-супертипа относятся только к одному из ее *подтипов*. Наличие в модели *подтипов* и *супертипов* усложняют проектирование и создают определенные трудности в реализации. Поэтому важно на ранней стадии проектирования установить, является ли наличие *супертипов* в модели необходимым.

установить, много ли одинаковых свойств имеют различные *подтипы*. Следует помнить, что чем меньше *подтипы* похожи друг на друга, тем больше вероятность введения *супертипа* ;

найти *экземпляр сущности*, который можно обоснованно включить в более чем один *подтип*. Поскольку это противоречит определению *супертипа*, предлагаемое разбиение недопустимо.

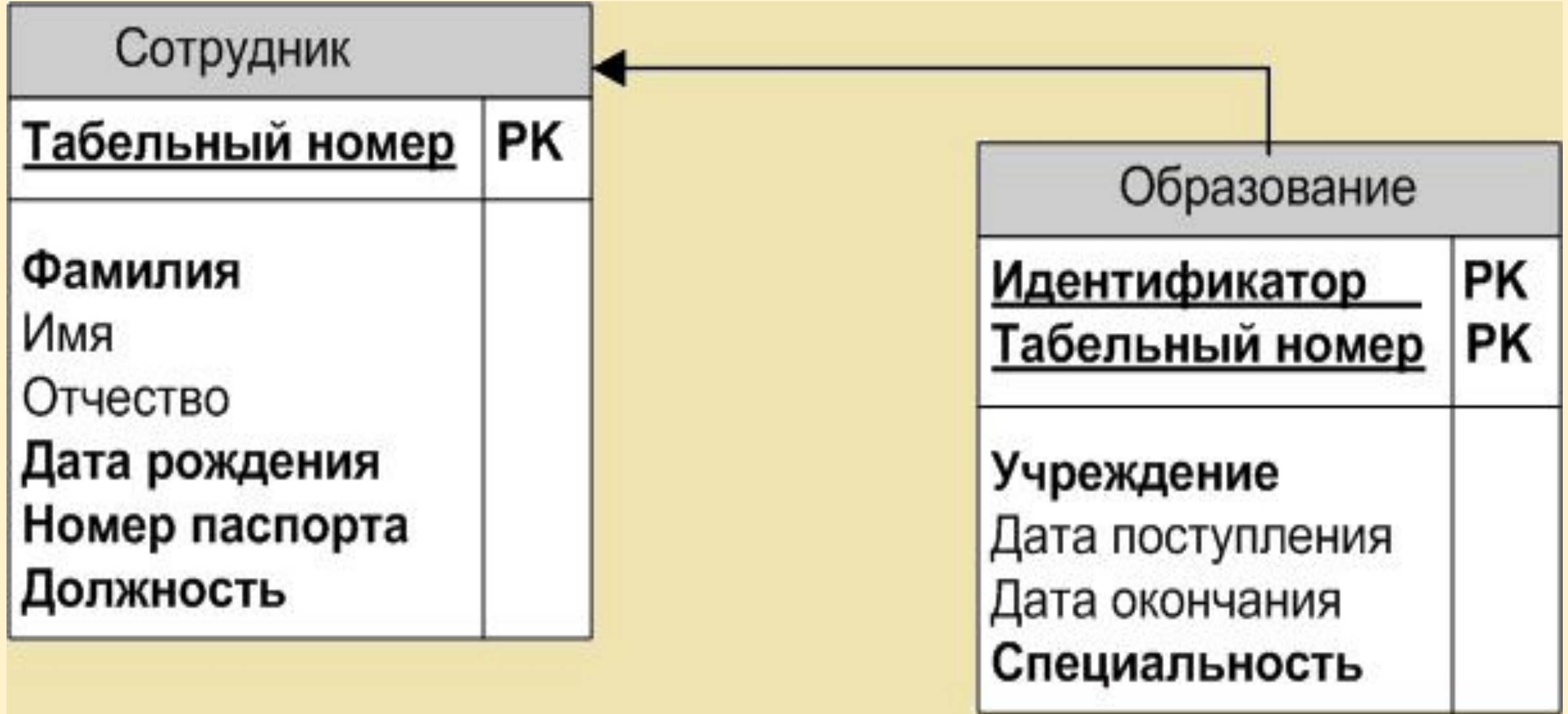
дополнить *потенциальный* ключ дополнительными атрибутами, то они не должны содержать нулевых значений.

Проектировщик должен тщательным образом изучить *домены* каждого атрибута с точки зрения их реализуемости в СУБД, с участием аналитиков внести в них изменения, если условие реализуемости не выполняется. При этом проектировщик руководствуется следующим:

для реализации
реляционного ХД
требуется использовать
реляционную или
объектно-реляционную
СУБД, например, MS SQL
Server

в большинстве
реляционных СУБД в
качестве языка
манипулирования и
описания данных
используется SQL,
поддерживающий
определенные стандарты,
например, ANSI SQL

Представление отношения между двумя сущностями на ER-диаграмме



Определение мощности связи отношения между сущностями "Сотрудник" и "Образование"

Сотрудник

<u>Табельный номер</u>	PK
Фамилия	
Имя	
Отчество	
Дата рождения	
Номер паспорта	
Должность	

Образование

<u>Идентификатор</u>	PK
<u>Табельный номер</u>	PK
Учреждение	
Дата поступления	
Дата окончания	
Специальность	

Свойства базы данных

Категории: Определение, Имя, Прочее, Действие ссылке

Мощность: 0 или более, 1 или более, 0 или 1, ровно 1, диапазон

Тип отношения: идентифицирующее, идентифицирующее

Наличие родительского объекта: обязательно, необязательно

Иерархическое отношение: 1 к 0 или более

Именованная связь между сущностями "Сотрудник" и "Образование"

Примеры_1.vsd - Microsoft Visio

Файл Правка Вид Вставка Формат Сервис Данные База данных Фигура Окно Справка

Arial 10пт

Фигуры

Поиск фигур:

Отношение сущности

- Родительская таблица и кат...
Перетащите фигуру на страницу, чтобы соединить родительскую...
- Категория
Перетащите фигуру на страницу, чтобы указать, что одна или не...
- Категория и дочерняя табл...
Перетащите фигуру на страницу, чтобы соединить категорию с д...
- Динамическая соединительн...
Перетащите фигуру на страницу, чтобы соединить два процесса.

Объекты реляционной базы данных

Страница-1

Свойства базы данных

Категории:

- Определение:
- Имя
- Прочее
- Действие ссылке

Вербальная фраза: имеет

Обратная фраза: относится к

Фонетическое имя: Каждый сотрудник имеет Образование

Заметки:

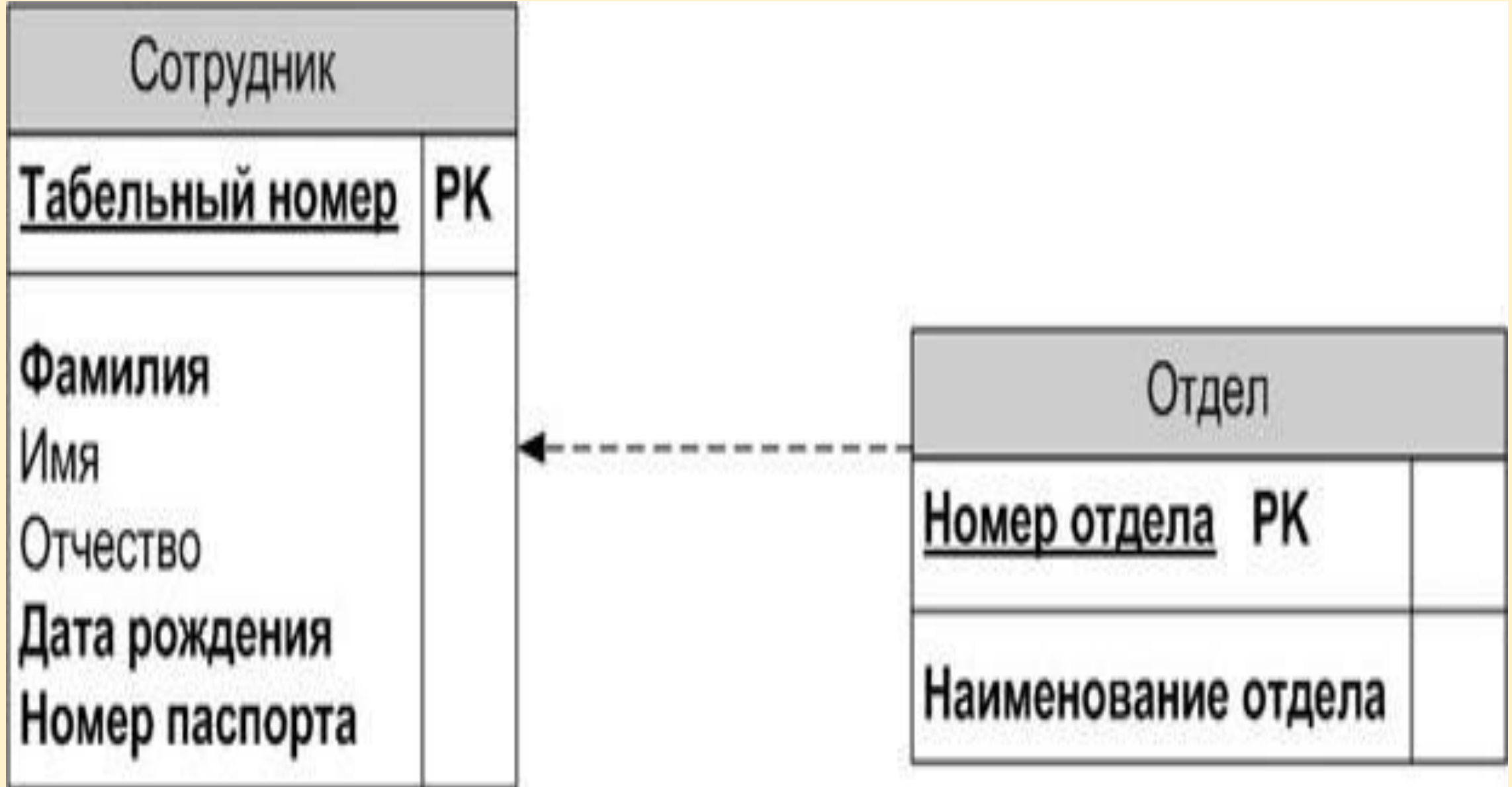
Начало = Освободить

Стр. 1/1

Сотрудник	
<u>Табельный номер</u>	PK
Фамилия	
Имя	
Отчество	
Дата рождения	
Номер паспорта	
Должность	

Образование	
<u>Идентификатор</u>	PK
<u>Табельный номер</u>	PK
Учреждение	
Дата поступления	
Дата окончания	
Специальность	

Неидентифицирующая связь



Связь "многие ко многим"



Характеристическая – зависимая дочерняя сущность, которая связана только с одной родительской и по смыслу хранит информацию о характеристиках родительской сущности.

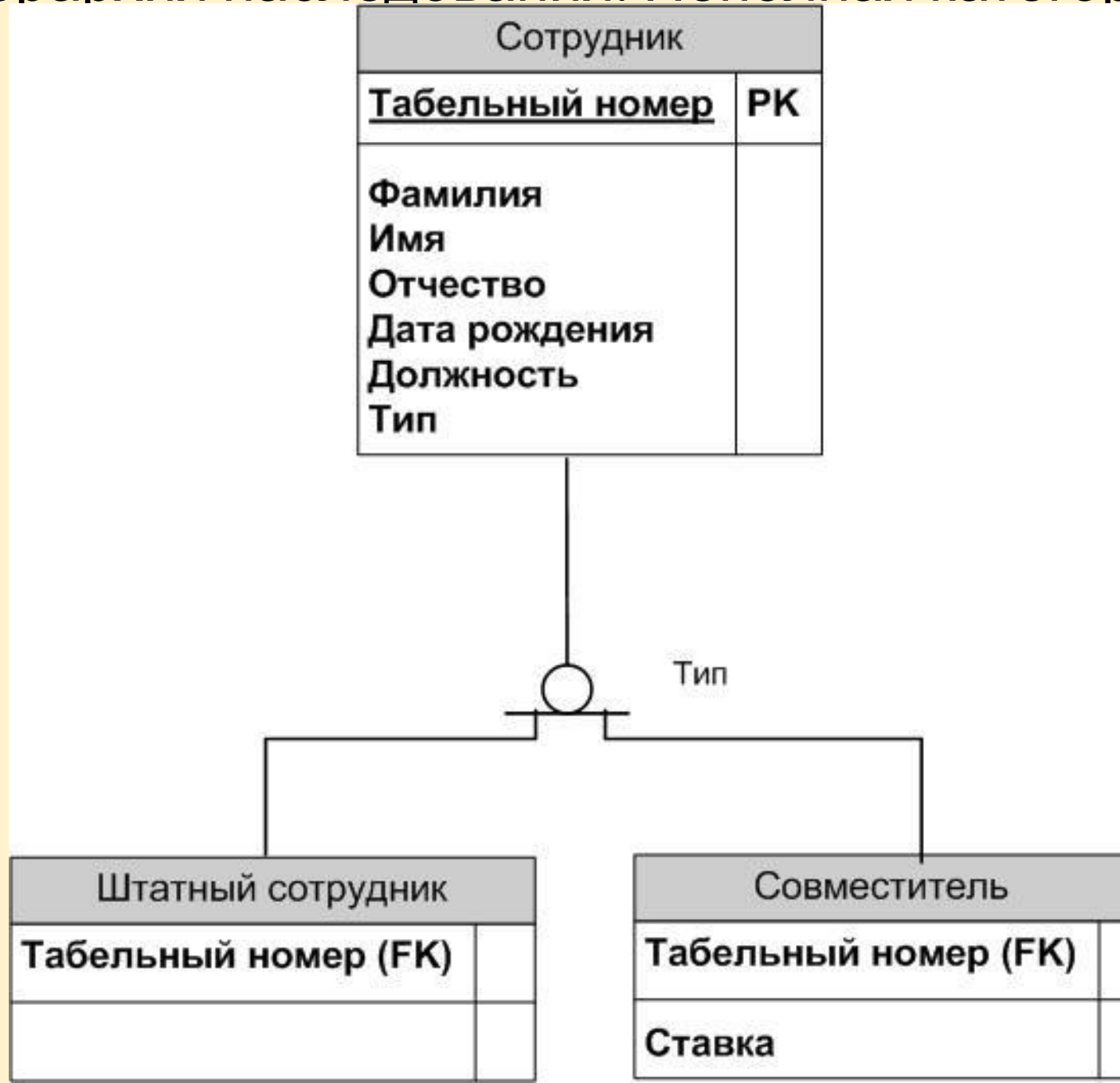
Ассоциативная – сущность, связанная с несколькими родительскими сущностями. Такая сущность содержит информацию о связях сущностей.

Именующая – частный случай ассоциативной сущности, не имеющей собственных атрибутов (только атрибуты родительских сущностей, мигрировавших в качестве внешнего ключа).

Категориальная – дочерняя сущность в иерархии наследования.

Иерархия наследования (*subtype relationship*), или иерархия категорий, представляет собой особый тип объединения сущностей, которые разделяют общие характеристики.

Иерархия наследования. Неполная категория

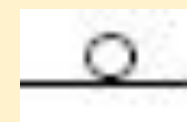
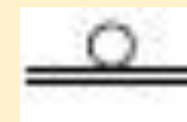


Иерархия наследования. Полная категория

Сотрудник	
<u>Табельный номер</u>	РК
Фамилия	
Имя	
Отчество	
Дата рождения	
Должность	
Тип	

Полная категория помечается
символом

неполна
я

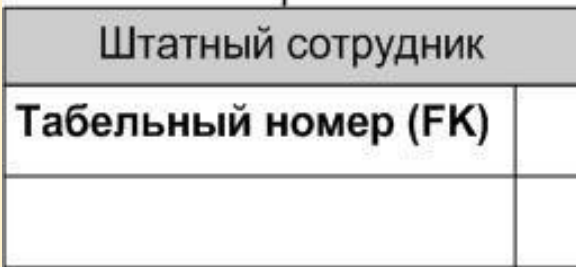
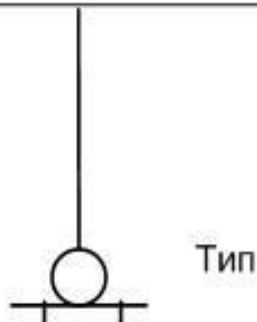


Определение сущностей с общими (по определению) атрибутами.

Штатный сотрудник	
Табельный номер	
Фамилия	
Имя	
Отчество	
Дата рождения	
Должность	

Совместитель	
Табельный номер (FK)	
Фамилия	
Имя	
Отчество	
Дата рождения	
Должность	
Ставка	

- Предположим, в процессе проектирования созданы сущности "Штатный сотрудник" и "Совместитель". Можно заметить, что часть атрибутов у этих сущностей (*Фамилия, Имя, Отчество, Дата рождения, Должность*) имеет одинаковый смысл.



- *Перенос общих атрибутов в сущность – родовой предок.* В случае обнаружения совпадающих по смыслу атрибутов следует создать новую сущность "Сотрудник" – родовой предок, и перенести в нее общие атрибуты (*Фамилия, Имя, Отчество, Дата рождения, Должность*).
- *Создание неполной структуры категорий.* Создается категориальная связь от новой сущности – родового предка к старым сущностям-потомкам. Новая сущность дополняется атрибутом – дискриминатором категории (*Тип*).

Консультант	
Табельный номер	
Фамилия	
Имя	
Отчество	
Дата рождения	
Организация	

- *Создание полной структуры категорий.* Проводится дополнительный поиск сущностей, имеющих общие по смыслу атрибуты с родовым предком. В примере это сущность "Консультант"

- *Комбинации полной и неполной структур категорий.* При необходимости создание иерархии категорий можно продолжить. Для каждого потомка может найтись сущность с общими атрибутами, тогда сущность-потомок становится родовым предком для новых потомков и т. д.

разделить сложные атрибуты на атомарные;

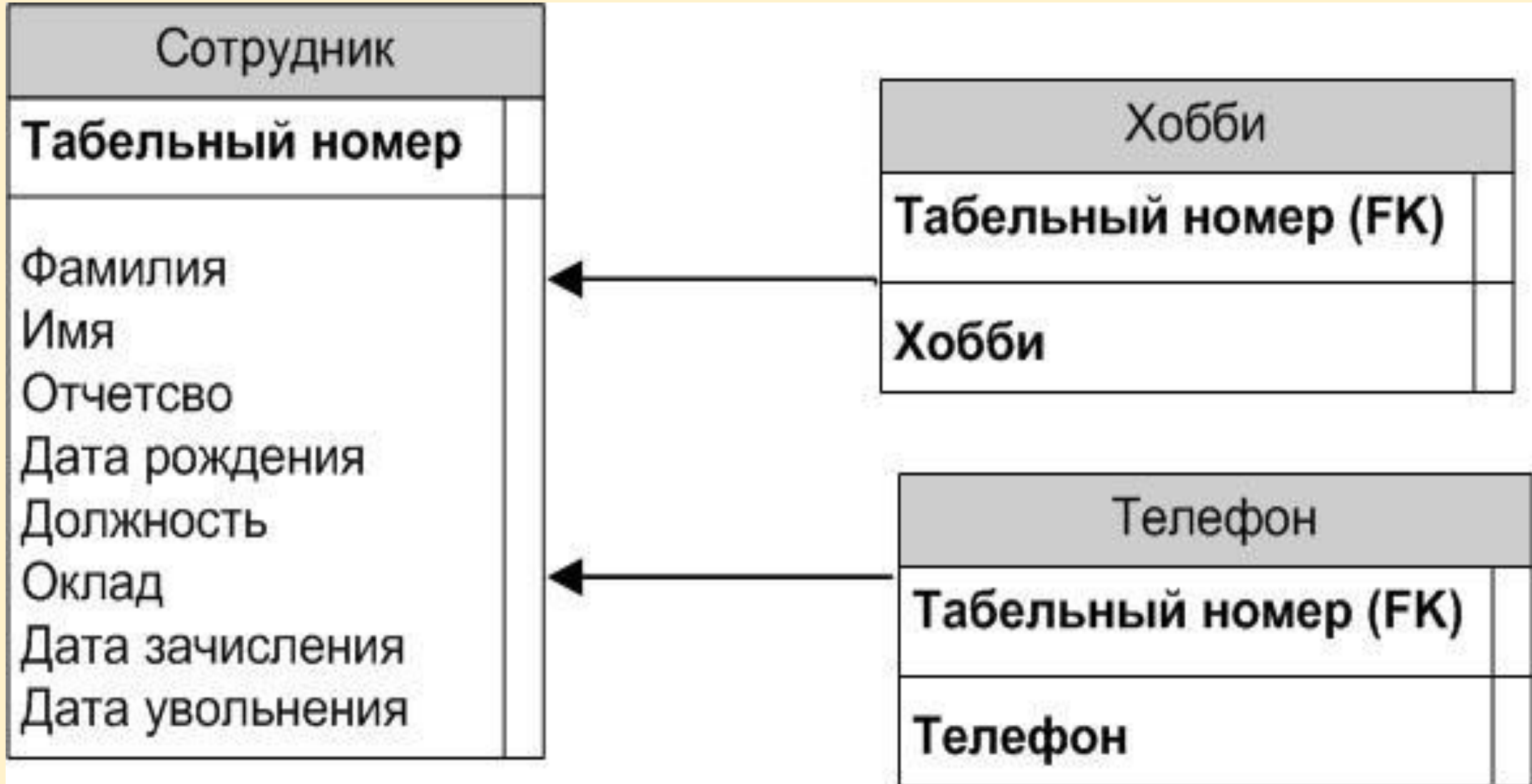
создать новую сущность;

перенести в нее все "повторяющиеся" атрибуты;

выбрать возможный ключ для нового РК (или создать новый РК);

установить идентифицирующую связь от прежней сущности к новой, РК прежней сущности станет внешним ключом (FK) для новой сущности.

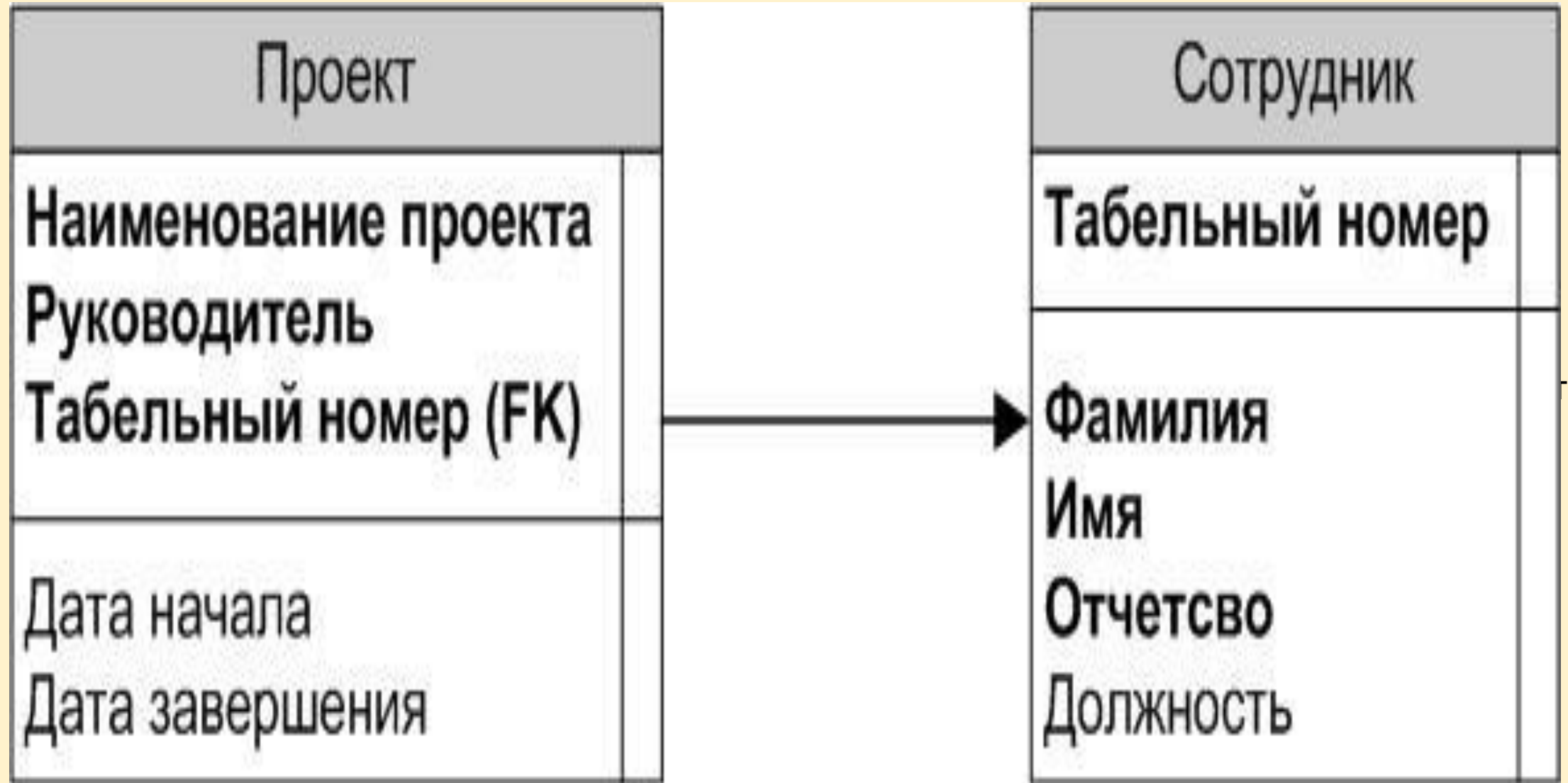
Сущность "Сотрудник", приведенная к первой нормальной форме



Проект	
Наименование проекта Табельный номер руководителя	
Дата начала Дата завершения Фамилия Имя Отчетсво Должность	

Сущность "Проект"

- *Вторая нормальная форма (2NF) Сущность находится во второй нормальной форме, если она находится в первой нормальной форме и каждый неключевой атрибут полностью зависит от первичного ключа (не должно быть зависимости от части ключа). Вторая нормальная форма имеет смысл только для сущностей, имеющих сложный первичный ключ.*



- *Вторая нормальная форма* позволяет избежать следующих аномалий при выполнении операций:
 - обновления (UPDATE). Происходит дублирование данных о сотруднике, если он руководит несколькими проектами. Если данные о сотруднике изменяются, необходимо менять несколько записей (по числу ведомых проектов);
 - вставки (INSERT). Невозможно ввести данные о сотруднике, если он в данный момент не руководит проектами;
 - удаления (DELETE). Если сотрудник временно прекращает руководство проектами, данные о нем теряются.



а;

- В *третьей нормальной форме* каждый *атрибут сущности* зависит от ключа, от всего ключа целиком и ни от чего другого, кроме как от ключа. *Третья нормальная форма* также позволяет избежать ряда аномалий.
- Обновление (UPDATE). Происходит дублирование данных об окладе, если должность занимают несколько сотрудников. Если оклад, соответствующий должности, меняется, необходимо менять несколько записей (по числу сотрудников на одной должности).
- Вставка (INSERT). Невозможно ввести данные об окладе, соответствующем должности, если в данный момент нет сотрудника, занимающего эту должность.
- Удаление (DELETE). В случае удаления из таблицы сотрудника, занимающего уникальную должность, данные об окладе теряются.

Четвертая нормальная форма (4NF) требует отсутствия многозначных зависимостей между атрибутами.

Преподаватель	
Табельный номер	
Предмет Группа	

Преподаватель-Предмет	
Табельный номер Предмет	

Преподаватель-Группа	
Табельный номер Группа	

Для каждой *сущности предметной области* базы данных необходимо:

получить список *атрибутов сущности*;

определить функциональные зависимости;

определить возможные ключи, в частности, рассмотрев *уникальный идентификатор сущности*;

выполнить нормализацию сущности;

назначить первичные ключи новых, полученных в результате нормализации сущностей;

назначить первичные ключи новых, полученных в результате нормализации сущностей;

Для каждой связи между сущностями необходимо:

определить мощность связи;

определить обязательность вхождения сущности в связь;

разрешить связи "многие ко многим";

назначить первичные ключи ассоциативных сущностей, исходя из уникального идентификатора связи и процедуры миграции ключей при нормализации;

определить неключевые атрибуты ассоциативных сущностей, если они необходимы;

сформировать бизнес-правила поддержки целостности связей;

документировать логическую модель реляционной базы данных;

проверить логическую модель реляционной базы данных.

***ЗАДАНИЕ
НА ПРАКТИЧЕСКОЕ ЗАНЯТИЕ***

- Требуется построить диаграмму «сущность-связь» для требуемой предметной области. Предметная область определяется преподавателем. Диаграмма должна содержать как минимум три сущности. Между сущностями должны быть реализованы необходимые связи.
- Отчет должен содержать требуемую диаграмму «сущность-связь».