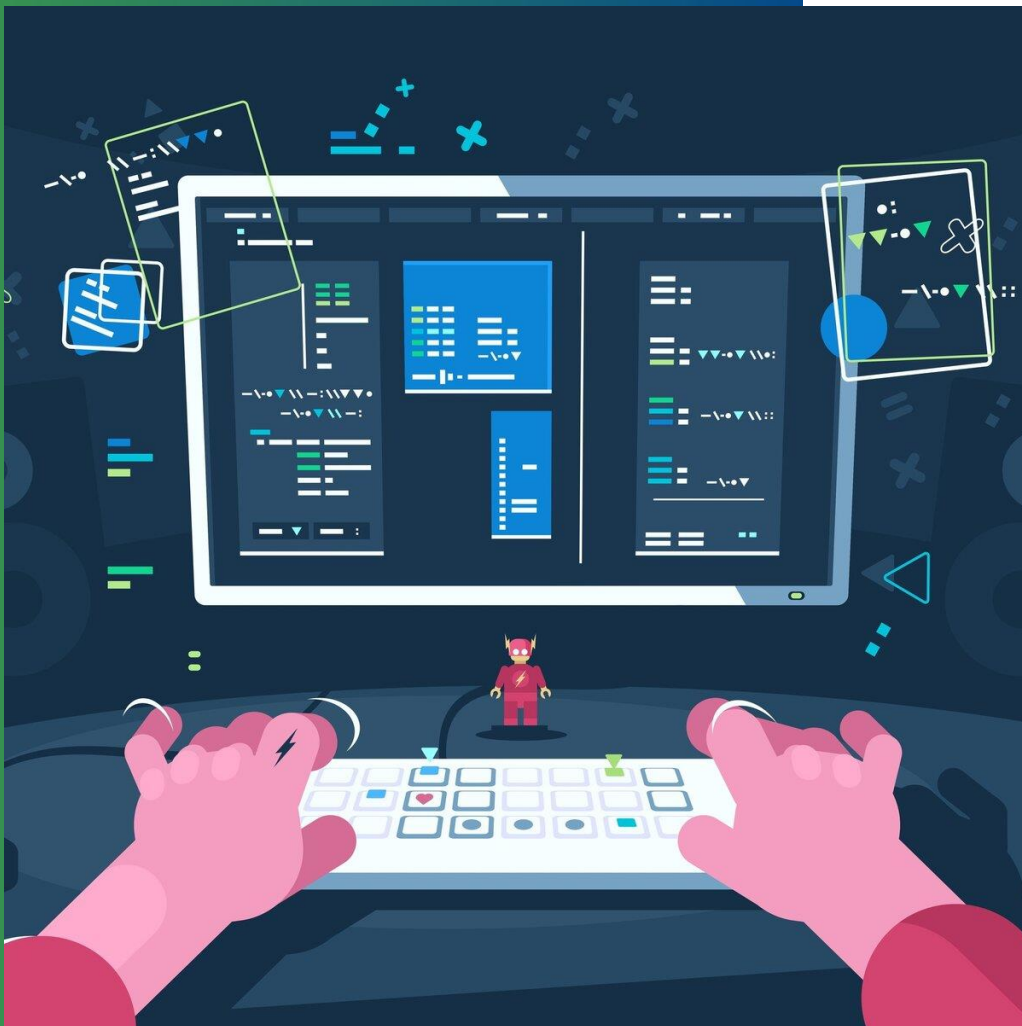


# Модель IDEF1X

Быстренина Ирина Евгеньевна,  
кандидат педагогических наук, доцент,  
доцент кафедры прикладной  
информатики РГАУ-МСХА имени К. А.  
Тимирязева



**Каким образом  
организованы  
данные в системе?**





Традиционно процедуру проектирования базы данных разбивают на три этапа, каждый из которых завершается созданием соответствующей информационной модели.

**Этап 1-й. Концептуальное проектирование** – создание схемы БД, включающего определение важнейших сущностей (таблиц) и связей между ними, но не зависящего от модели БД (иерархической, сетевой, реляционной и т. д.) и физической реализации (целевой СУБД).

**Этап 2-й. Логическое проектирование** – развитие концептуальной схемы БД с учетом принимаемой модели (иерархической, сетевой, реляционной и т.д.).

**Этап 3-й. Физическое проектирование** – развитие логической схемы БД с учетом выбранной целевой СУБД.



## *ERD*

*(Entity – Relationship Diagrams,  
«сущность–связь»)*

Впервые предложена  
П. Ченом в 1976 г.

## *IDEF1X*

Основана на  
методологии IDEF1,  
разработанной Т.  
Рэмеем.

В 1981 г. принята ICAM в качестве  
федерального стандарта США.



# I. Последовательность шагов при концептуальном проектировании

---

## 1. Выделение сущностей.

В графической нотации IDEF1X для отображения сущности используются обозначения, изображенные на следующем рисунке.

Участки



а) независимая

Пути



б) зависимая



# I. Последовательность шагов при концептуальном проектировании

## 2. Определение атрибутов.



У независимой сущности «Участки» в качестве первичного ключа назначен суррогатный ключ, у зависимой сущности «План» – первичный ключ составной, состоящий из пяти атрибутов.



# I. Последовательность шагов при концептуальном проектировании

---

## 2. Определение атрибутов.

**Ключ** – один или несколько атрибутов сущности, служащих для однозначной идентификации ее экземпляров, их быстрого поиска или задания связи между экземплярами родительской и дочерней сущностей.

- **суперключ (superkey)** – атрибут или множество атрибутов, которое единственным образом идентифицирует экземпляр сущности.
- **потенциальный ключ (potential key)** – суперключ, который не содержит подмножества, также являющегося суперключом данной сущности, т. е. суперключ, содержащий минимально необходимый набор атрибутов, единственным образом идентифицирующих экземпляр сущности.
- **первичный ключ (primary key)** – потенциальный ключ, который выбран для уникальной идентификации экземпляров внутри сущности;
- **альтернативные ключи (alternative key)** – потенциальные ключи, которые не выбраны в качестве первичного ключа.

В дочерней сущности для моделирования связи должен присутствовать набор атрибутов, соответствующий первичному ключу родительской. Этот набор атрибутов в дочерней сущности принято называть **внешним ключом (foreign key)**.



# I. Последовательность шагов при концептуальном проектировании

## 3. Определение связей.

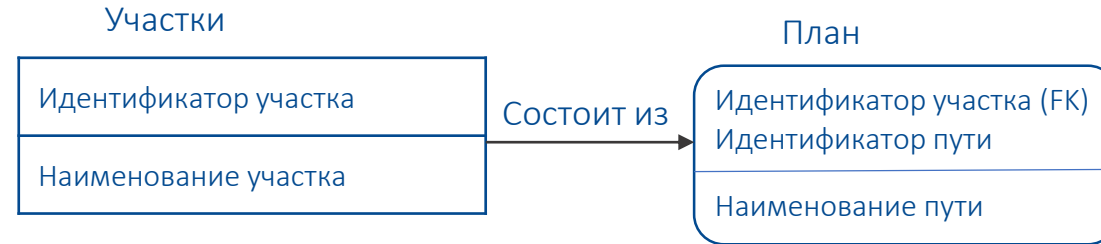
Внешний вид	Тип и обязательность связи	Мощность связи справа
	Обязательная, идентифицирующая	1
	Обязательная, идентифицирующая	0 .. ∞
 <i>Z</i>	Обязательная, идентифицирующая	0 или 1
 <i>P</i>	Обязательная, идентифицирующая	1 .. ∞
 <число>	Обязательная, идентифицирующая	<число>
	Обязательная, неидентифицирующая	0 .. ∞
	Необязательная, неидентифицирующая	0 .. ∞



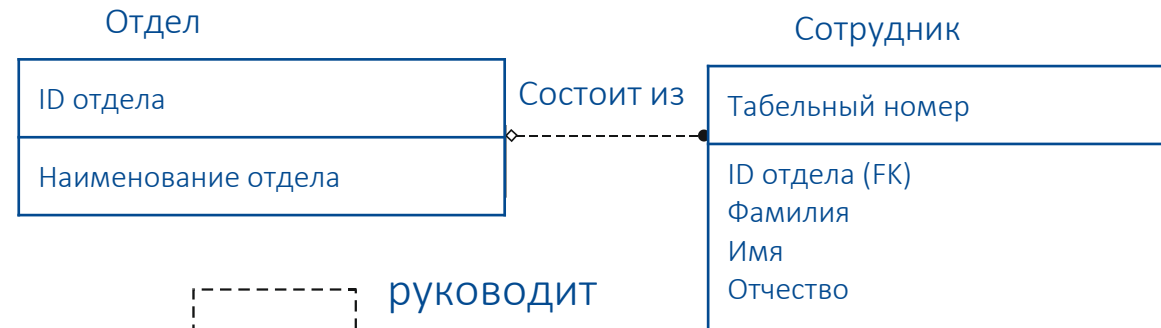


# Примеры связей

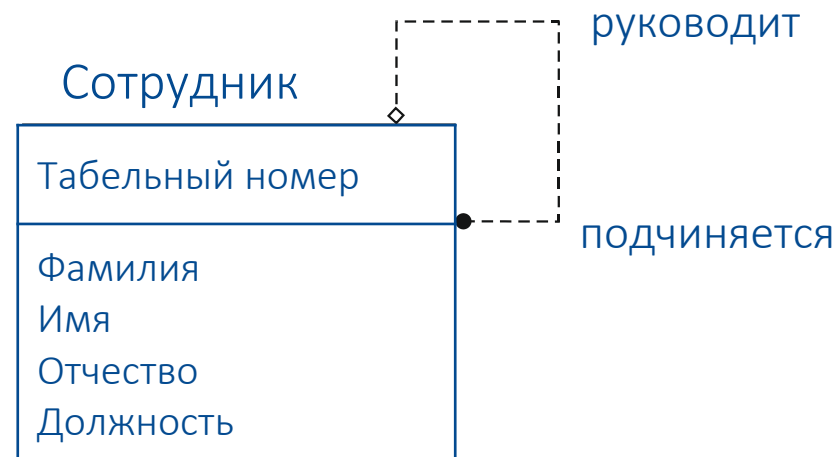
а) идентифицирующая



б) неидентифицирующая



в) рекурсивная

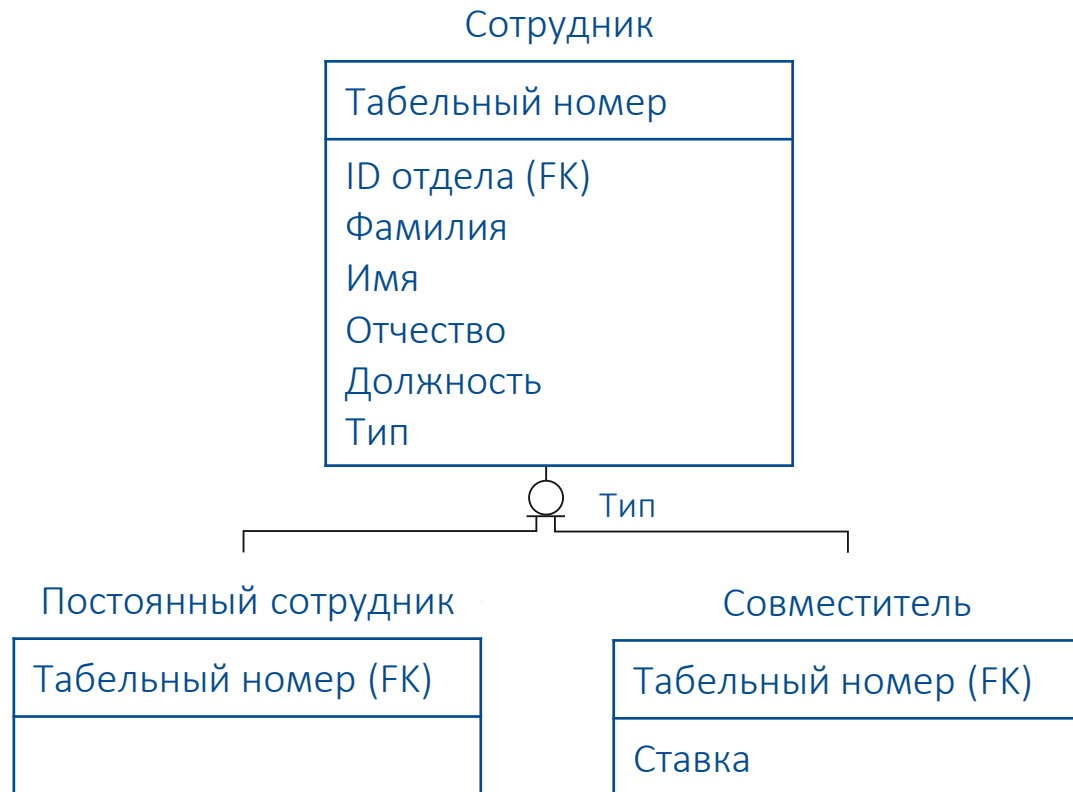




# I. Последовательность шагов при концептуальном проектировании

## 4. Определение суперклассов и подклассов.

Суперкласс – сущность, включающая в себя подклассы.

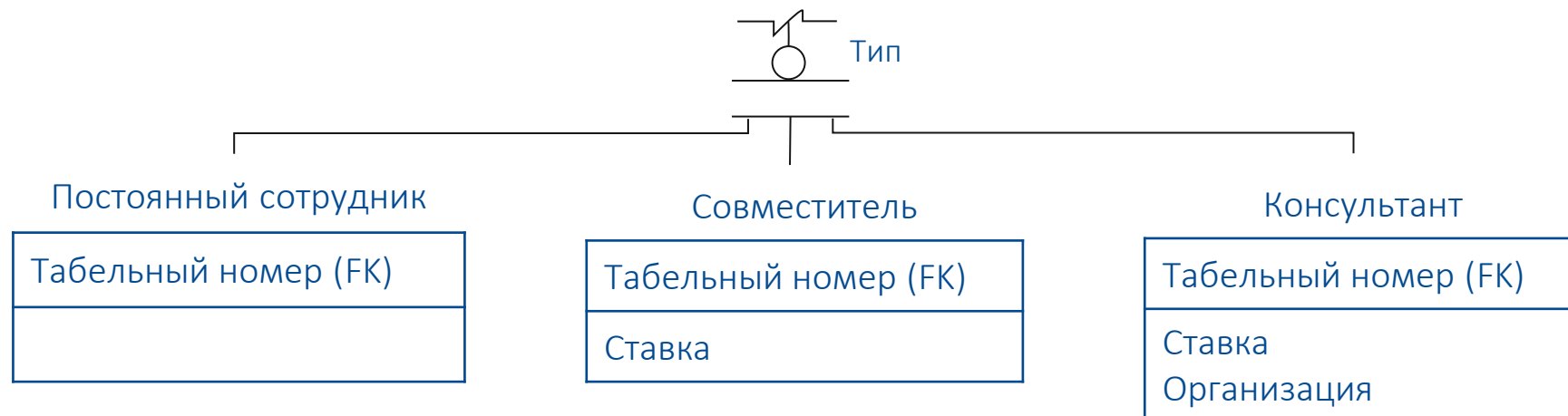




# I. Последовательность шагов при концептуальном проектировании

## 4. Определение суперклассов и подклассов.

Для каждой категории требуется указать **дискриминатор** – атрибут родового предка, который показывает, как отличить одну сущность от другой. В приведенном примере дискриминатор – атрибут «Тип».

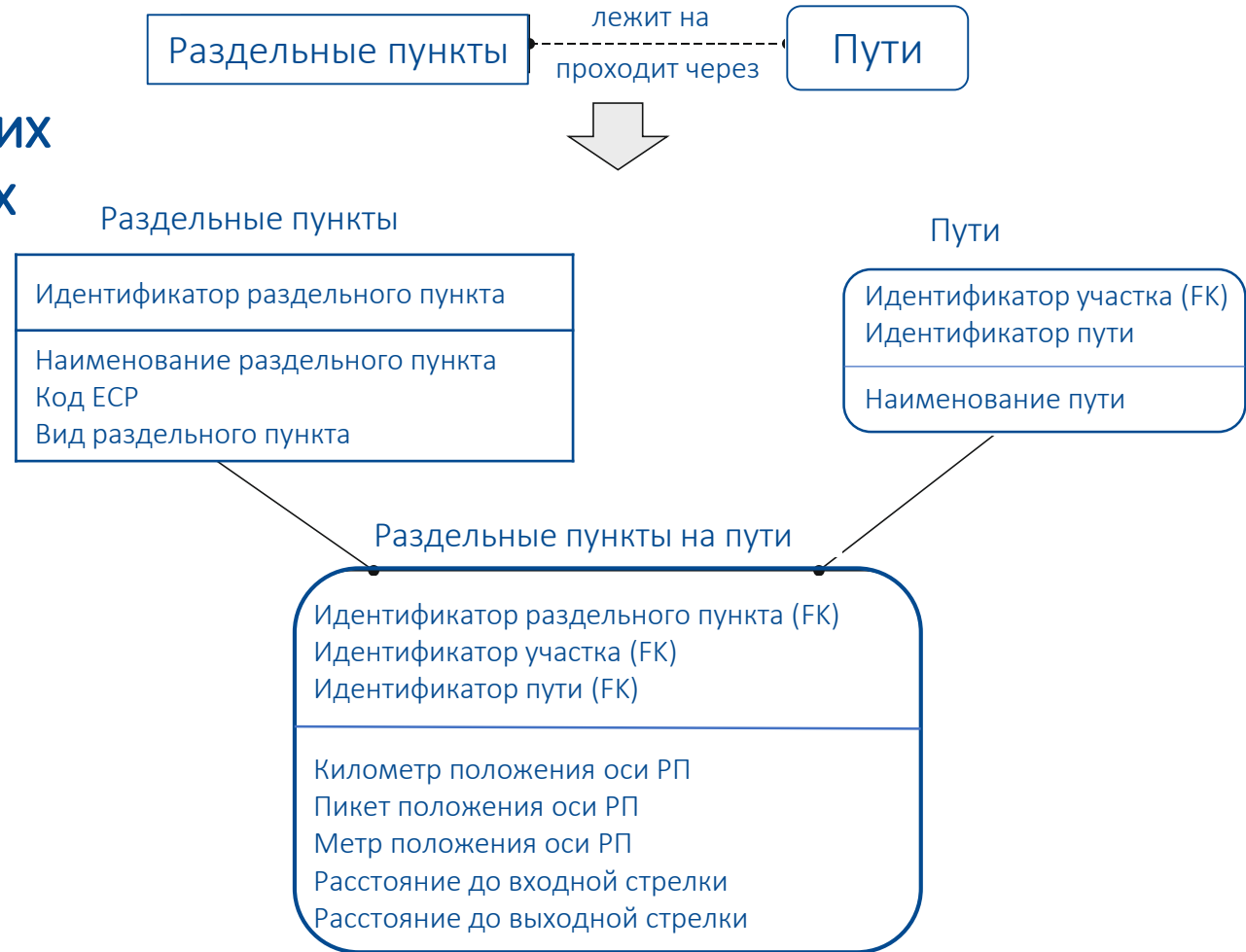




## II. Логическое проектирование с использованием методологии IDEF1X

### 1. Удаление и проверка элементов, не отвечающих принятой модели данных

#### 1.1. Удаление связей N:M.





## II. Логическое проектирование с использованием методологии IDEF1X

---

1. Удаление и проверка элементов, не отвечающих принятой модели данных.

1.2. Удаление связей с атрибутами.

Связи с атрибутами должны быть преобразованы в сущности.

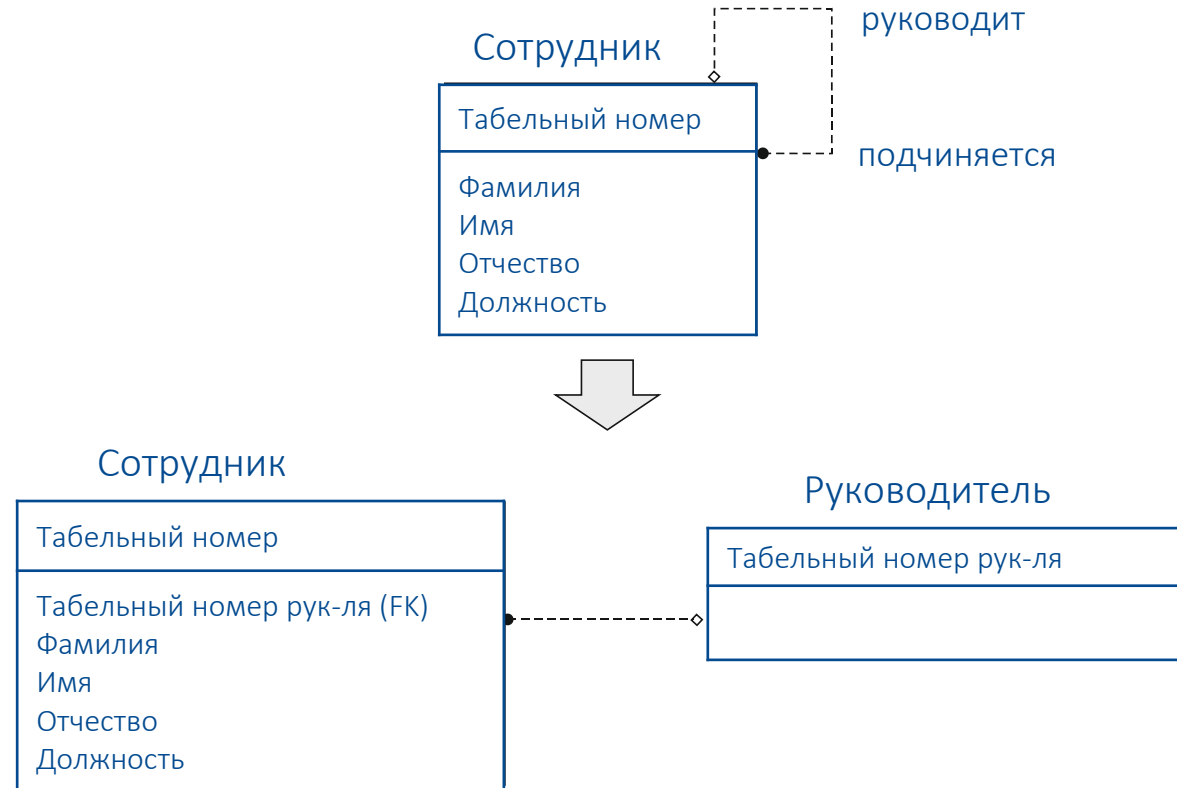
1.3. Удаление сложных связей (со степенью участия более 2).

Сложную связь заменяют необходимым количеством бинарных связей 1:N со вновь созданной сущностью, которая и показывает эту связь.



## II. Логическое проектирование с использованием методологии IDEF1X

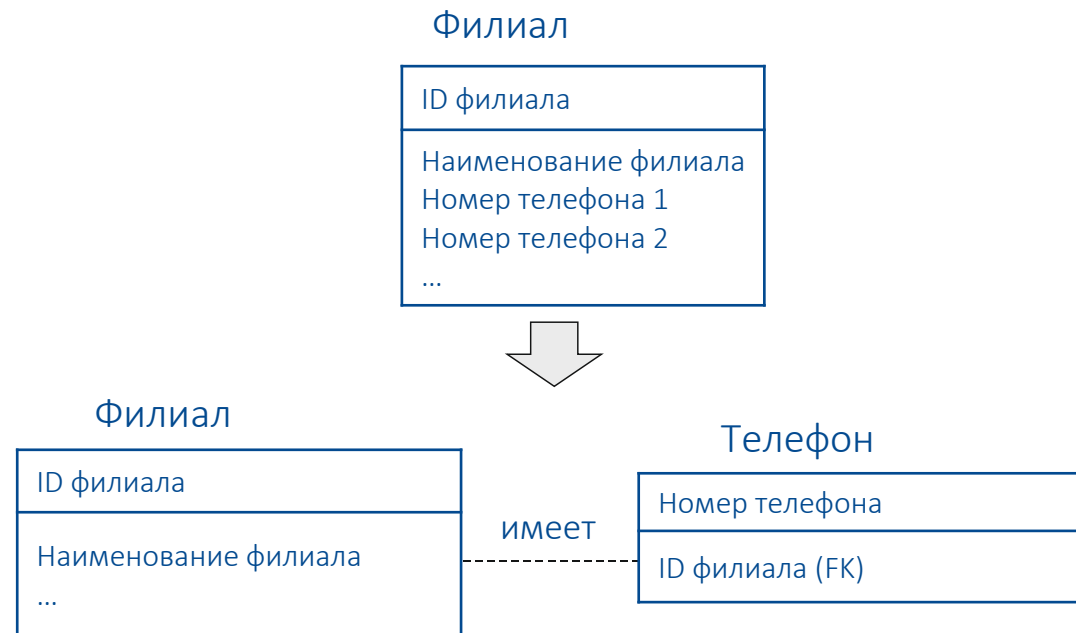
1. Удаление и проверка элементов, не отвечающих принятой модели данных.
  - 1.4. Удаление рекурсивных связей (со степенью участия 1).





## II. Логическое проектирование с использованием методологии IDEF1X

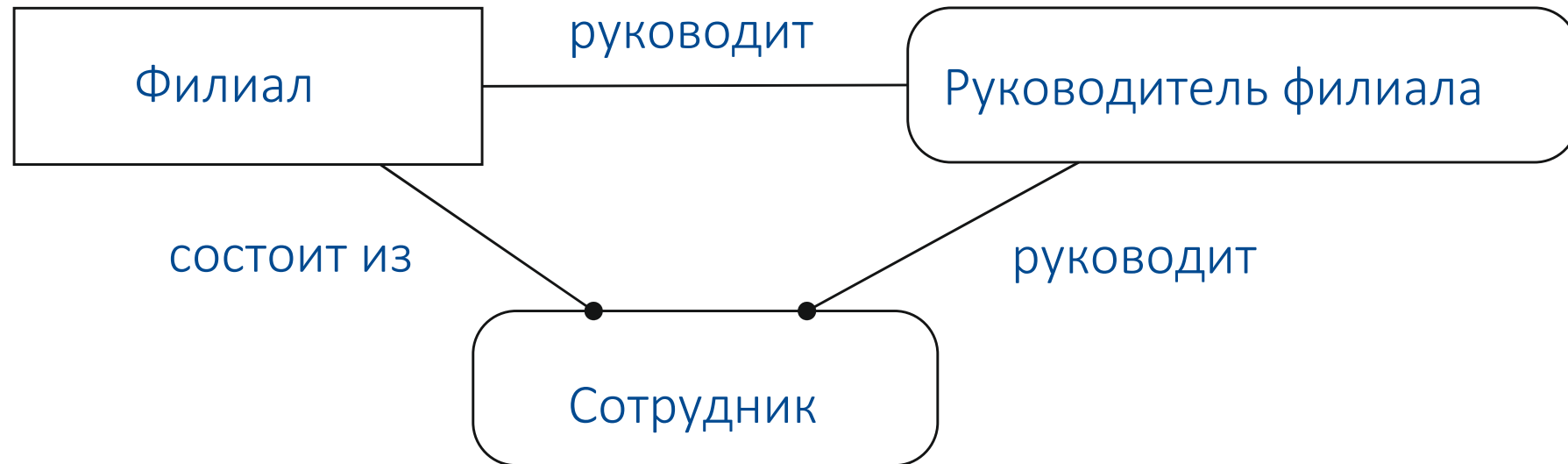
1. Удаление и проверка элементов, не отвечающих принятой модели данных.
  - 1.5. Удаление многозначных атрибутов (атрибутов имеющих несколько значений).





## II. Логическое проектирование с использованием методологии IDEF1X

1. Удаление и проверка элементов, не отвечающих принятой модели данных.
  - 1.6. Удаление избыточных связей.



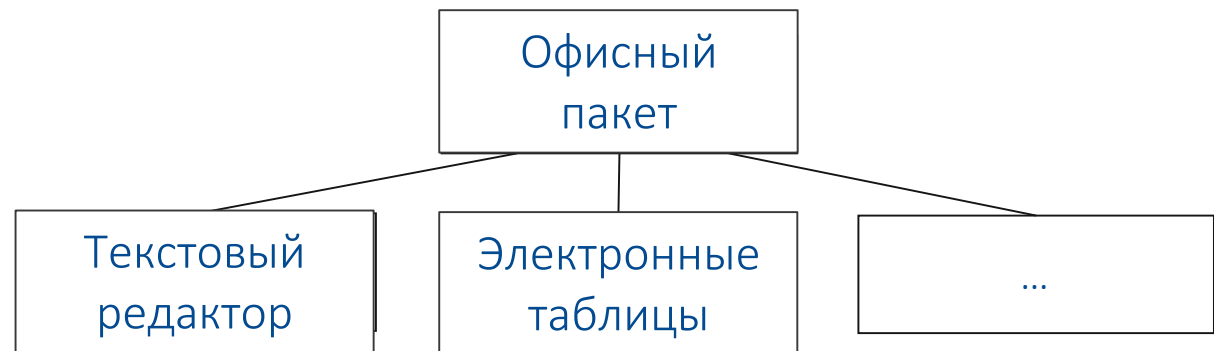
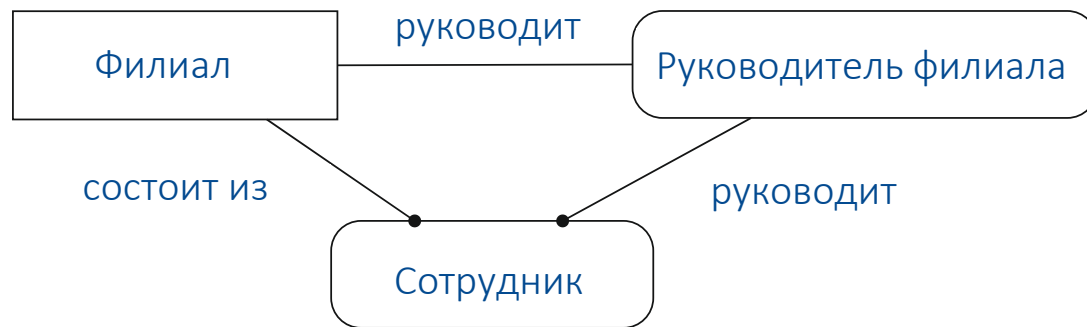




## II. Логическое проектирование с использованием методологии IDEF1X

1. Удаление и проверка элементов, не отвечающих принятой модели данных.

1.7. Перепроверка связей 1:1.





## II. Логическое проектирование с использованием методологии IDEF1X

---

### 2. Проверка модели с помощью правил нормализации.

Основная идея нормализации заключается в том, чтобы каждый факт хранился в одном месте, т. е. чтобы не было дублирования данных. Многие из требований нормализации, как правило, уже учитываются при выполнении предыдущих шагов проектирования.

Процесс нормализации впервые был предложен Э.Ф. Коддом в 1972 г. Сначала было предложено три вида нормальных форм (1NF, 2NF и 3NF). Затем Р. Бойсом и Э.Ф. Коддом (1974 г.) было сформулировано более строгое определение третьей нормальной формы, которое получило название нормальная форма Бойса–Кодда (BCNF). Вслед за BCNF появились определения четвертой (4NF) и пятой (5NF или PJNF) нормальных форм (Р. Фагин, 1977 и 1979 г.)



## II. Логическое проектирование с использованием методологии IDEF1X

---

### 3. Определение требований поддержки целостности данных.

Ограничения целостности данных представляют собой ограничения, которые вводятся с целью предотвращения помещения в базу противоречивых данных.

К этим ограничениям относятся: обязательные данные, домены, бизнес-правила (бизнес-ограничения), ссылочная целостность.

Ссылочная целостность может быть обеспечена за счет использования триггеров.

**Триггер** – это хранимая в БД процедура, исполняемая СУБД автоматически при удалении (DELETE), вставке (INSERT) или обновлении (UPDATE) записи.

### III. Физическое проектирование с использованием методологии IDEF1X

---



Цель физического проектирования – преобразование логической схемы с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

Рекомендации:

1. Анализ необходимости введения контролируемой избыточности.
  - 1.1. Использование производных данных.
  - 1.2. Дублирование атрибутов.
2. Перенос логической схемы данных в среду целевой СУБД.
3. Реализация бизнес-правил и анализ транзакций.
4. Разработка механизмов защиты.
5. Организация мониторинга и настройка функционирования системы.



---

**Спасибо за внимание!**