

Проектирование БД

Модель разработки ИС

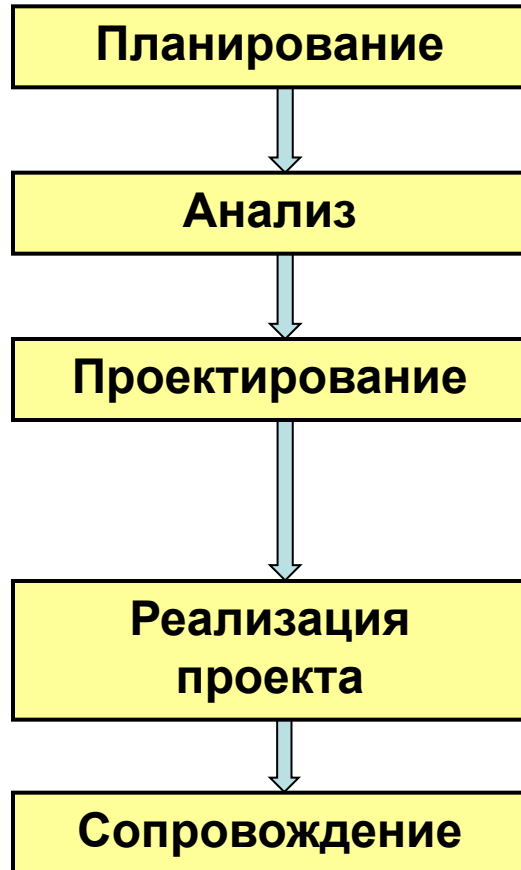
Процесс создания ИС называется разработкой системы

Технический подход к разработки ИС основан на модели жизненного цикла ИС

Жизненный цикл ИС - это история создания ИС

Жизненный цикл ИС состоит из ряда этапов

Этапы жизненного цикла ИС



Первоначальная оценка
Изучение реализуемости

Анализ требований пользователя
Оценка существующих систем
Выбор системной архитектуры

Спецификация требований
Разработка архитектуры системы
Разработка проекта БД
Разработка проекта приложений

Кодирование и отладка
Тестирование
Установка и настройка

Обслуживание
Оценка
Расширение

Модель разработки ИС

Две основных модели жизненного цикла ИС:

итерационная

каскадная

Итерационная модель жизненного цикла

Перекрывание во времени

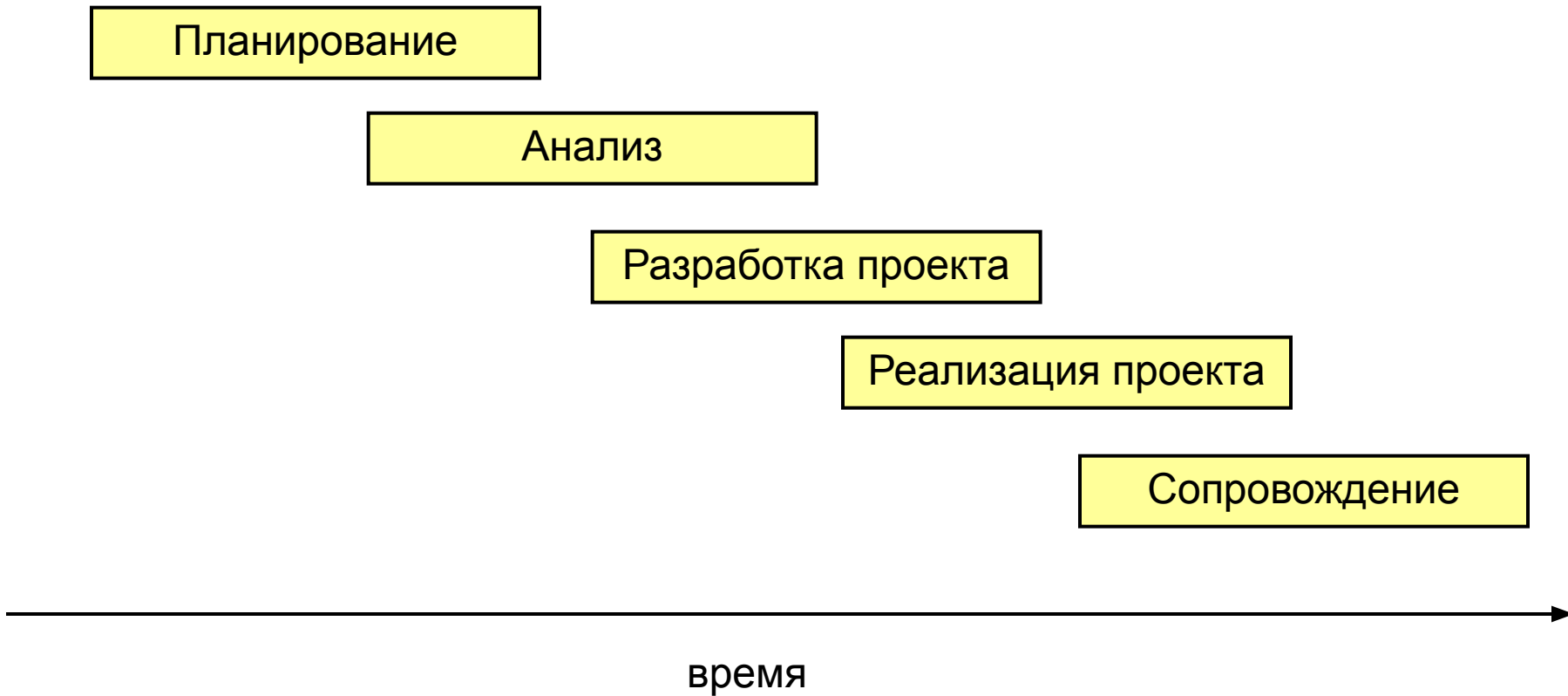


График плотности работ

Каскадная модель жизненного цикла

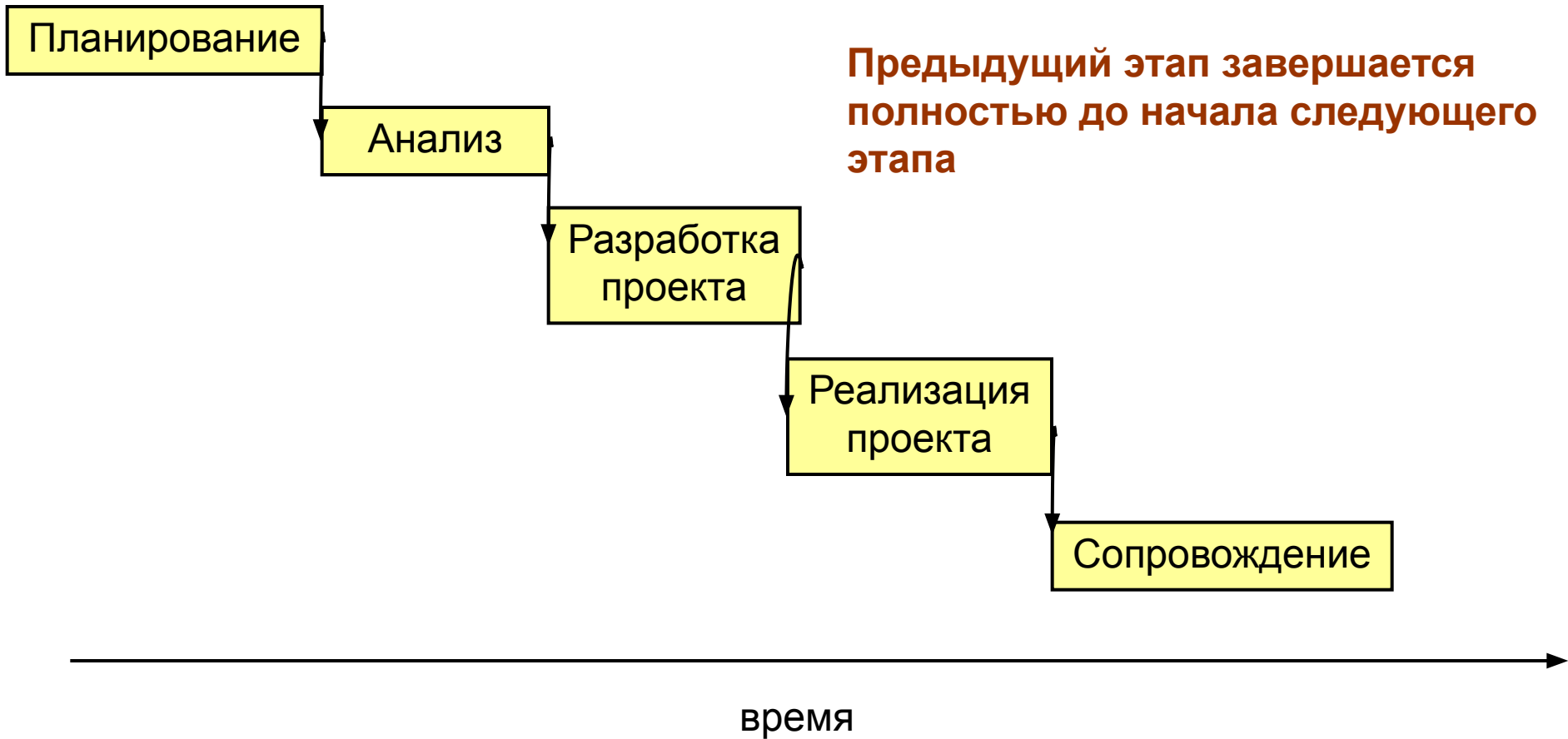


График рисков

Методы разработки ПО

На основных моделях жизненного цикла разработано множество методологий разработки ПО

Rational Unified Process (RUP) (*рациональный унифицированный процесс*)

V- Model

Spiral model (*спиральная модель*)

Test-driven development (TDD) (*разработка через тестирование*)

Lean Software Development (*гибкая разработка ПО*):

Extreme programming (*экстремальное программирование*);

Scrum (*срам*);

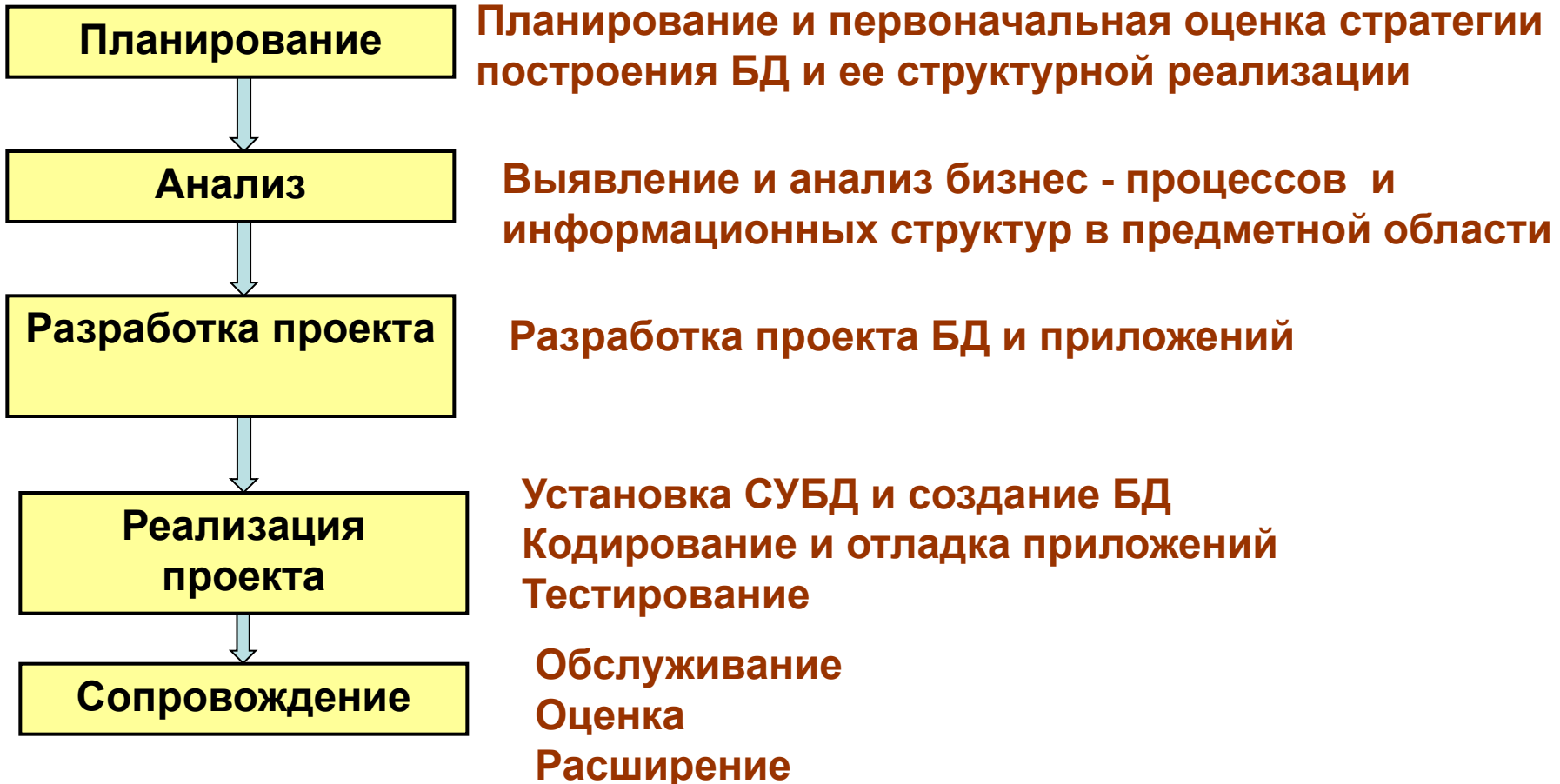
Kanban

...

Жизненный цикл БД

Процесс создания БД проходит в рамках процесса создания ИС

БД имеет собственный жизненный цикл



Проектирование БД

Проектирование БД – это процесс создание проекта БД

Процесс проектирования заканчивается созданием проекта

Основой проекта реляционной БД является **схема БД**, содержащая **набор взаимосвязанных отношений**, в которых определены все атрибуты, их типы и ограничения целостности, заданы первичные и вторичные ключи, определены индексы.

Проект БД **также** содержит схему физического размещения компонентов БД, описание **спецификации** реализуемых программных компонентов (запросы или представления, ХП, Т и т.п.) и др.

Проектирование БД

Основная проблема, которая решается при проектировании БД – это устранение избыточности данных, которые приводят к усложнению алгоритмов обработки данных и **аномалиям БД**, разрушающим целостность данных .

Различают *неизбыточное* и *избыточное* дублирование. Неизбыточное допускается, а избыточное - может приводить к аномалиям.

При попытке изменить состояние базы данных для некоторых схем отношений могут возникнуть нежелательные эффекты, которые и получили название **аномалий**.

Аномалии – это противоречие между предметной областью и данными, содержащимися в БД или сложности обработки данных.

Аномалии возникают из-за того, что наши знания о предметной области оказываются, по каким-то причинам, неправильно выражены в схеме БД или входящими в противоречие с ней.

Аномалии БД

Классические аномалии БД:

аномалии добавления

Проявляться в невозможности добавить к отношению требуемый кортеж (при добавлении нарушается ограничение целостности, поддерживаемое СУБД), например, для отношения

R(группа, специальность, предмет, лк, пз, преподаватель)

Причина аномалии - хранение в одном отношении разнородной информации (и о предмете, и о группе, и о специальности).

аномалии удаления

Проявляться, когда при удалении кортежа "теряется" полезная информация например, в отношении

R(группа, староста, студент, успеваемость)

удаление группы потребует удаление всех студентов

аномалии модификации

Проявляться, когда при изменении данных получить можно неоднозначную информацию об объекте, например, при замене старосты в отношении

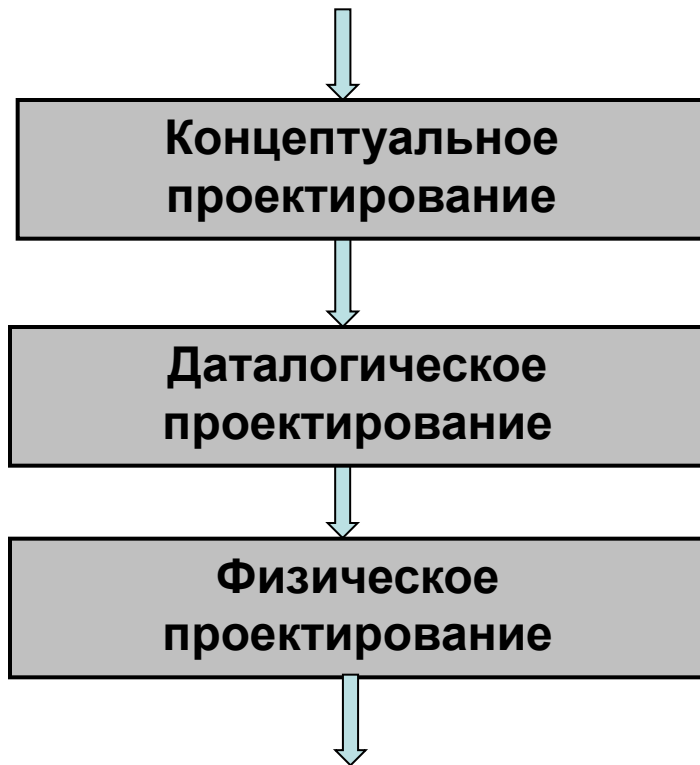
R(группа, староста, студент, успеваемость)

придется делать многократные замены, в результате у одной группы могут быть разные старосты...

Проектирование БД

Процесс проектирования БД представляет собой последовательность переходов от неформального описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели

В процессе проектирования выделяют следующие этапы



Исходным материалом для этапа проектирования БД является, полученная после этапа анализа, **бизнес-модель предметной области**, содержащая описание деятельности участников информационного процесса и информационные атрибуты этой деятельности (входные и выходные документы), и спецификация требований к системе, где описаны информационные компоненты системы, типы пользователей, их функции в системе, требования к масштабированию и быстродействию системы при выполнении запросов, и пр., пр.

Проектирование БД

Пример описания бизнес - модели предметной области на UML
(диаграмма бизнес-функций)

Use Case | **Use Case Specification for проверка наличия ...** ? X

General | Diagrams | Relations | Files

Name: | Name: | Package: Use Case View

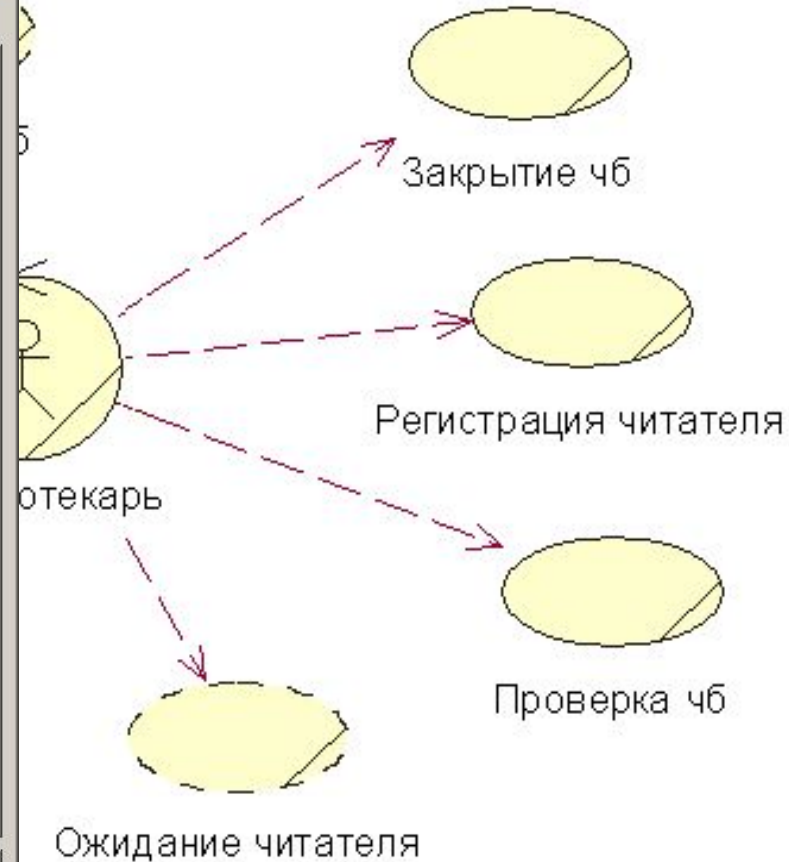
Stereotype: | Abstract

Rank: | Abstract

Documentation:

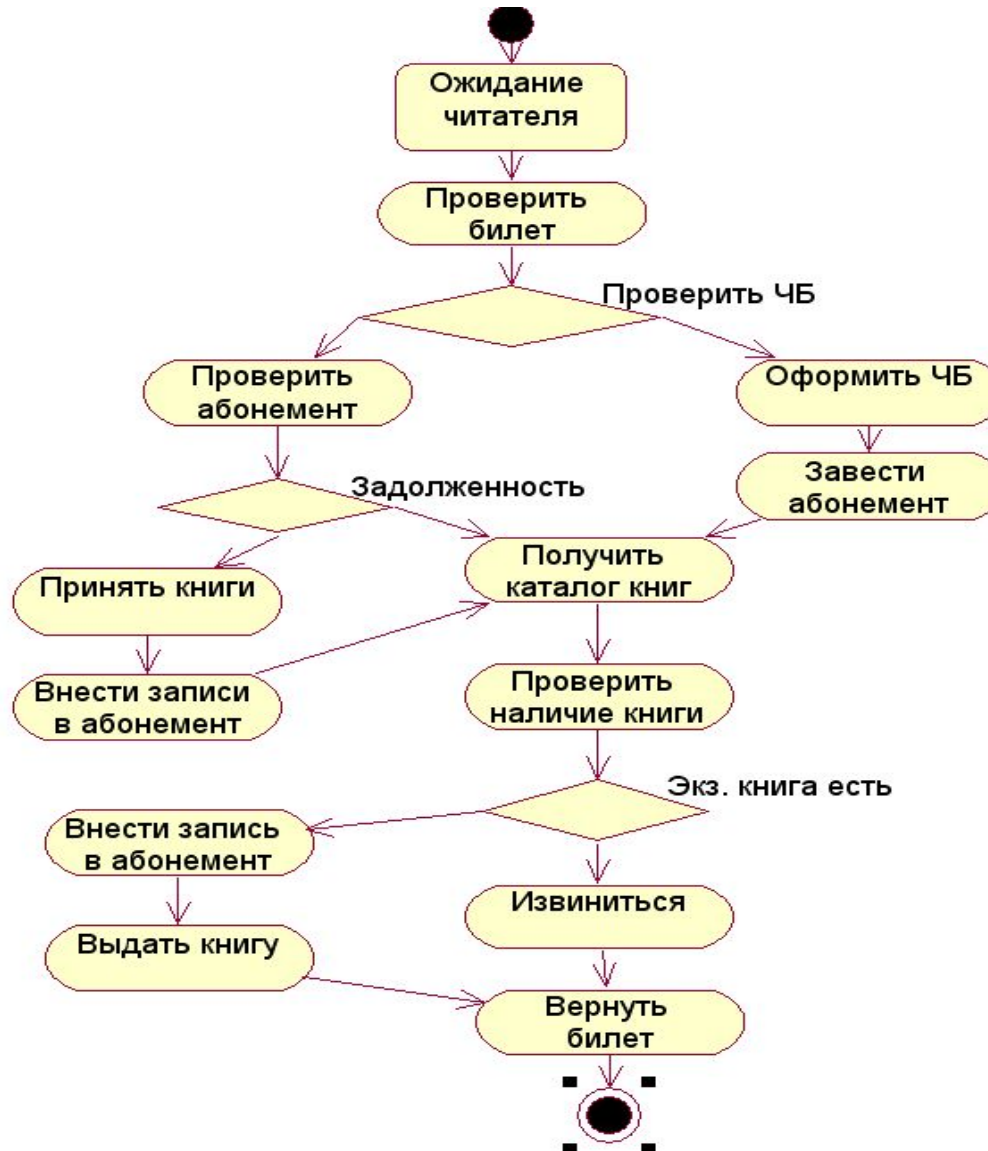
Documentation:

OK | Cancel | Apply | Browse | Help



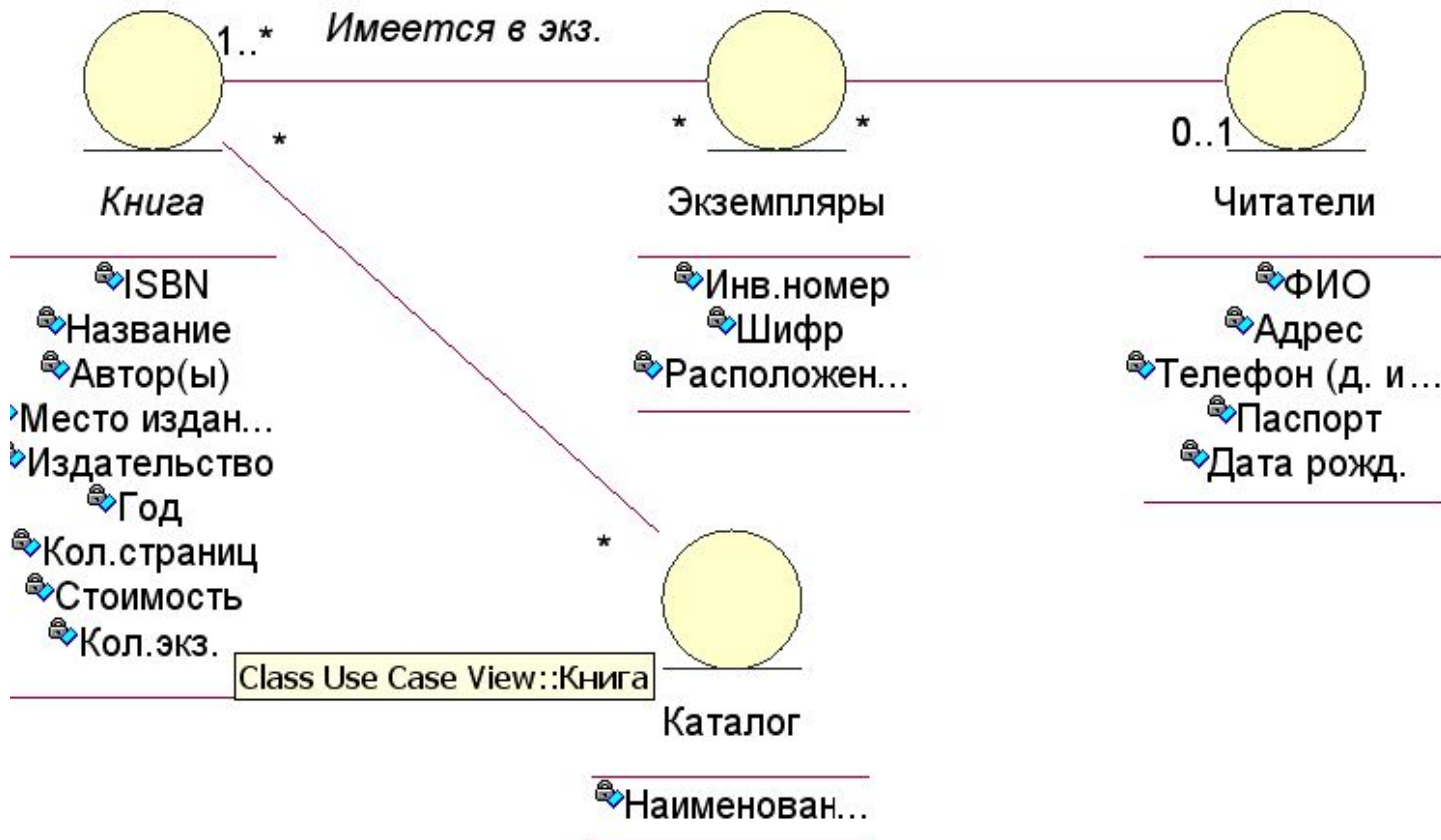
Проектирование БД

Пример описания бизнес - модели предметной области на UML
(диаграмма деятельности)

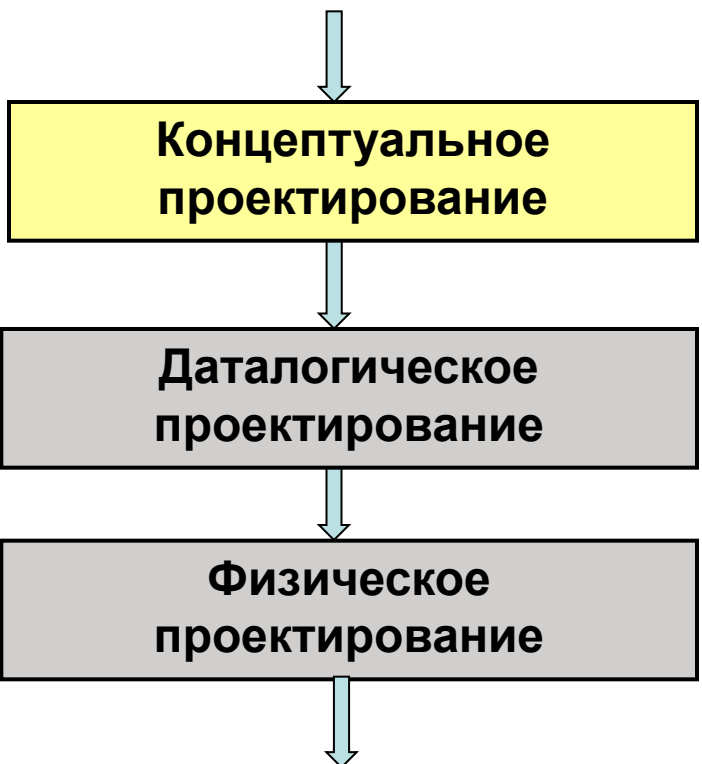


Проектирование БД

Пример описания бизнес - модели предметной области на UML
(диаграмма сущностей)



Этапы проектирования БД



```
graph TD; A[Концептуальное проектирование] --> B[Даталогическое проектирование]; B --> C[Физическое проектирование]; C --> D[ ];
```

Концептуальное проектирование

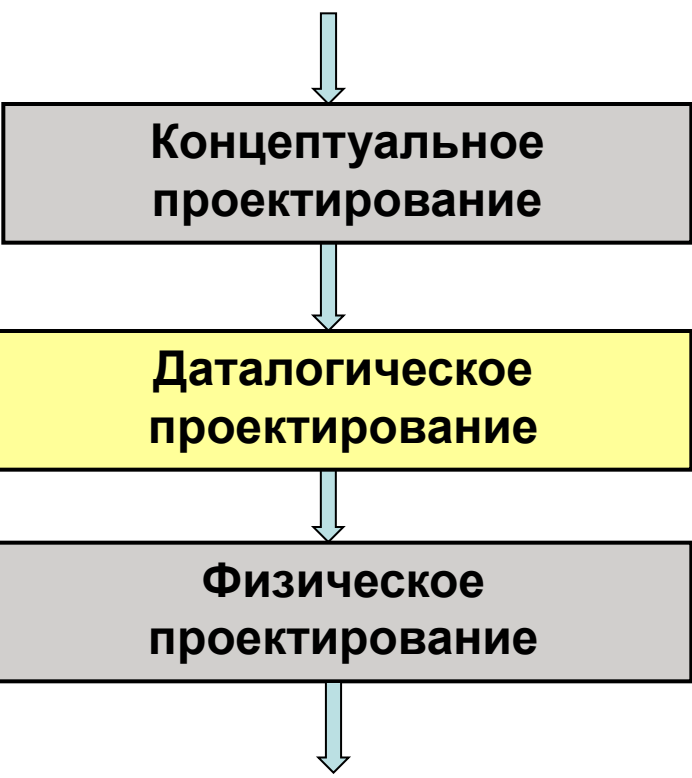
Даталогическое проектирование

Физическое проектирование

На основании выявленных информационных компонентов предметной области **строится инфологическая модель**, включающая объекты (сущности) предметной области и их атрибуты и

выполняется моделирование выполнения бизнес- функций всех участников, определенных на этапе разработки спецификации требований.

Этапы проектирования БД



```
graph TD; A[Концептуальное проектирование] --> B[Даталогическое проектирование]; B --> C[Физическое проектирование];
```

Концептуальное проектирование

Даталогическое проектирование

Физическое проектирование

Строится схема базы данных на основании инфологической модели или выявленных информационных атрибутов деятельности.

Этапы проектирования БД

```
graph TD; A[Концептуальное проектирование] --> B[Даталогическое проектирование]; B --> C[Физическое проектирование]; C --> D[ ]; style C fill:#ffff00
```

Концептуальное проектирование

Даталогическое проектирование

Физическое проектирование

Определение структур хранения БД в ОС и методов доступа к ним и моделей защиты БД.

Исходным материалом для этапа проектирования БД является схема БД, полученная на предыдущем этапе, и проект **внешней памяти** сервера БД, разработанный на этапе проектирования ИС, держащий структуру системы дисковых устройств...

Концептуальное проектирование

Результатом этого этапа проектирования является построение первичной информационной структуры базы данных, которая называется **концептуальной схемой базы данных** или **инфологической моделью**.

Концептуальная схема базы данных содержит сгруппированные атрибуты предметной области по признакам функциональной зависимости.

Основная цель этого класса моделей – моделирование выполнимости предъявленных к ИС функциональных требований.

Особенность инфологических моделей

- 1. Семантическая (смысловая) наполняемость**
- 2. Не зависимость от конкретной СУБД**

Концептуальное проектирование

Для описания **инфологических моделей** существует несколько типов такого рода моделей, например,

семантическая модель Хаммера – Мак-Леона

функциональная модель Шипмана

сущностная модель Чена (ER-модель)

UML - диаграммы и др.

На базе этих моделей строятся системы автоматизированного проектирования, так наз. **CASE- системы**.

На базе модели Чена созданы **ERWin, POWER DESIGNER** и др.

На базе модели UML создана **RATIONAL ROSE, PARADIGM PLUS, SELECT ENTERPRISE** и др.

Концептуальное проектирование

Основные функции CASE- систем

Создавать графические диаграммы для описания предметной области

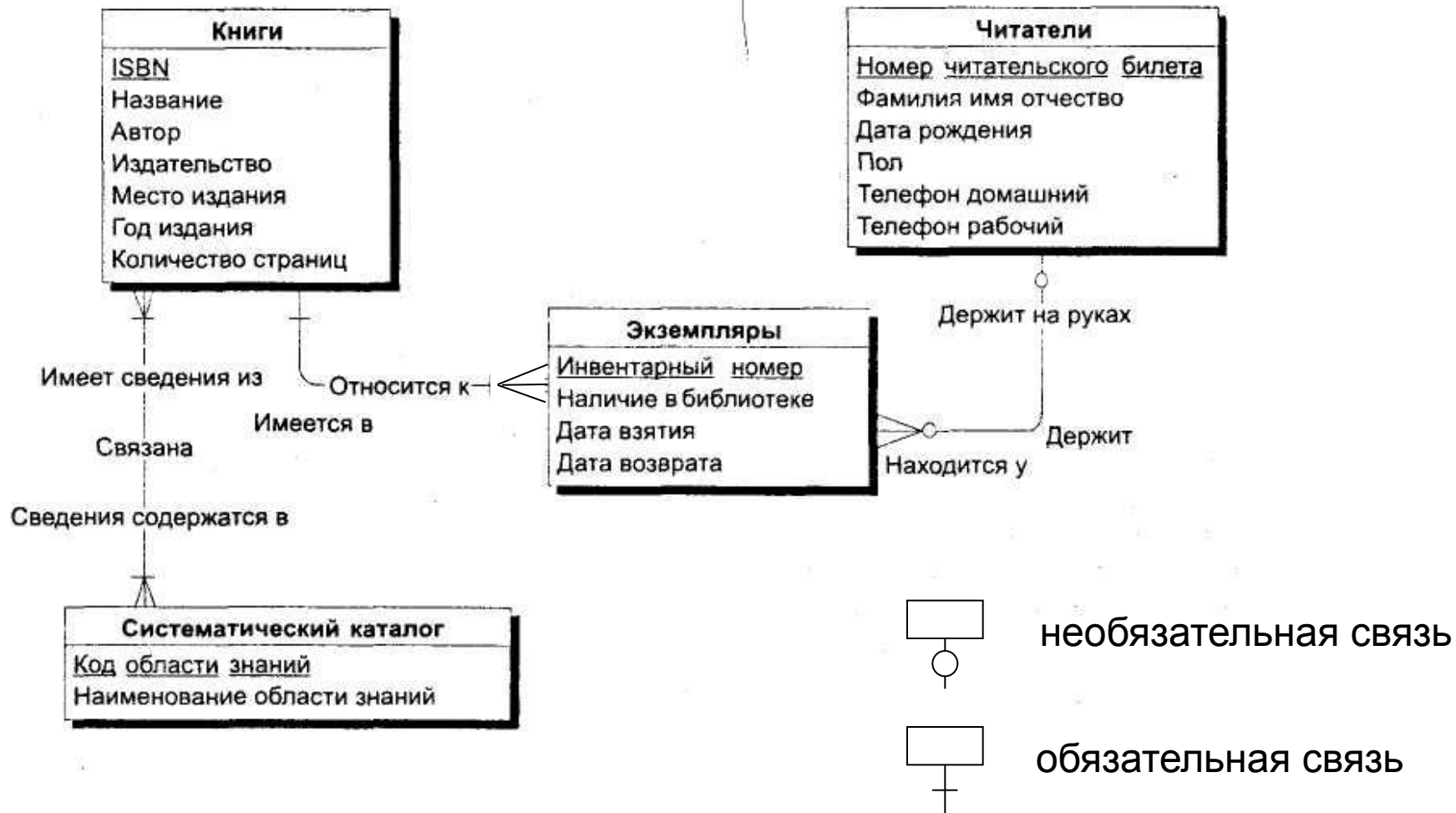
Выявлять логические ошибки в описании диаграмм

Создавать документацию и чертежи проекта

Генерировать программы по созданию структур для конкретной инструментальной среды

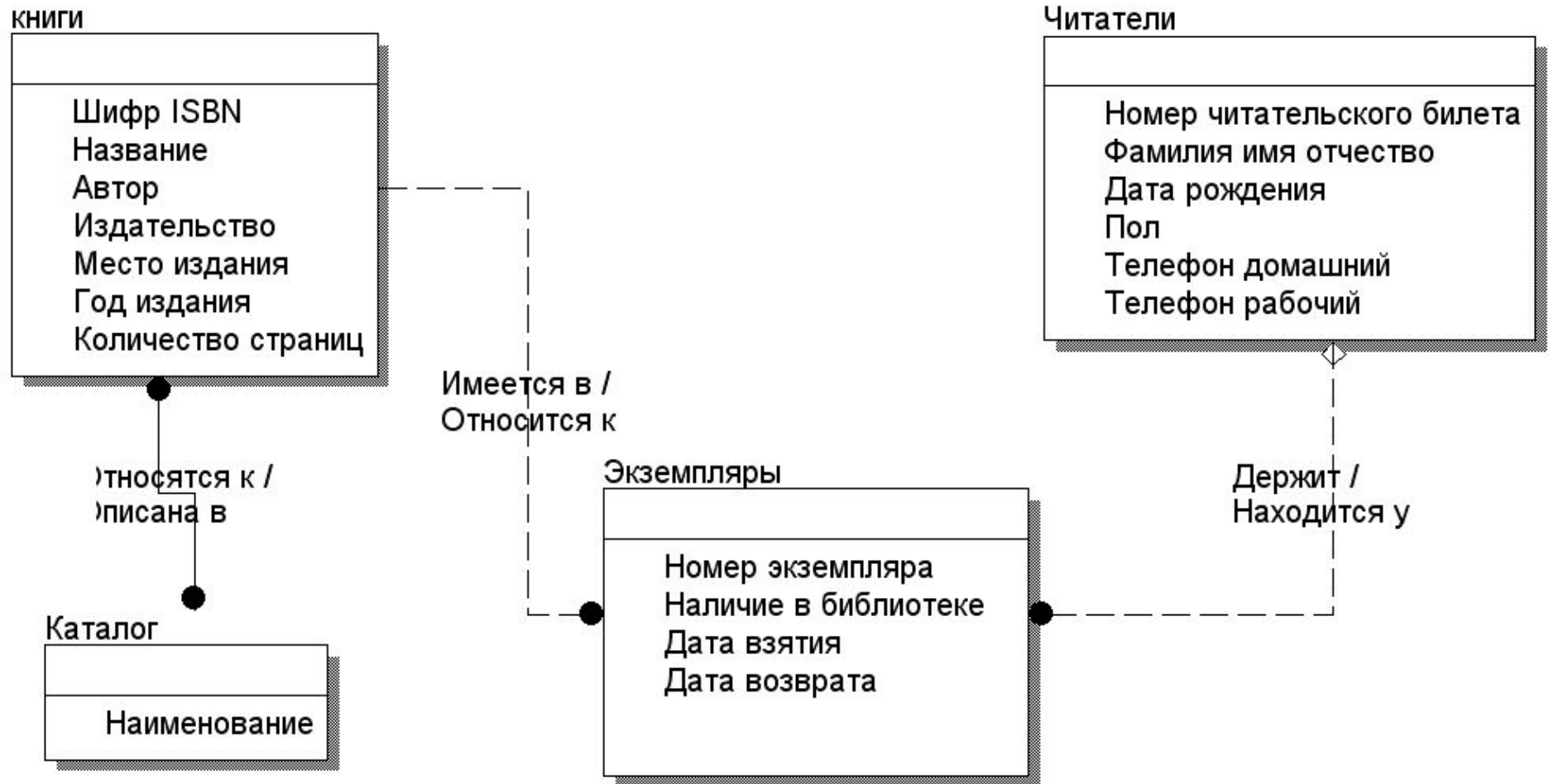
Концептуальное проектирование

Пример ER-модели данных ИС «библиотека» в нотации IE (POWER DESIGNER)



Концептуальное проектирование

Пример ER-модели данных ИС «библиотека» в нотации IDEF1X (ERWin)



Концептуальное проектирование

Описание ER-модели данных в нотации IDEF1X (**ERWin**)

ER-модели состоит из

сущностей

абстракция некоторого множества предметов реального мира, имеющих одни и те же характеристики, правила и поведения

атрибутов

абстракция характеристики, которой обладают все возможные экземпляры сущности

связей

абстракция некоторых отношений, которые систематически возникают между различными видами предметов реального мира

Концептуальное проектирование

Описание ER-модели данных в нотации IDEF1X (ERWin)

Отображение элементов ER-модели

сущность

прямоугольник с указанием названия в виде существительного

атрибутов

внутри сущности, к которой относится, с указанием имени в виде существительного

связей

линия, соединяющая две сущности с указанием отношения в виде глагола действия



Концептуальное проектирование

Описание ER-модели данных в нотации IDEF1X (**ERWin**)

Отображение элементов ER-модели

Сущности могут быть 2-х типов:

Независимая сущность

Экземпляры независимой сущности могут быть уникально идентифицированы без определения ее связей с другими сущностями;

Зависимая сущность

Зависимая сущность, наоборот, не может быть уникально идентифицирована без определения ее связей с другими сущностями.

Экземпляры

Экземпляры
Номер экземпляра
Наличие в библиотеке
Дата взятия
Дата возврата

Экземпляры

Экземпляры
Номер экземпляра
Наличие в библиотеке
Дата взятия
Дата возврата

Концептуальное проектирование

Описание ER-модели данных в нотации IDEF1X (ERWin)

Отображение элементов ER-модели

Связи могут 2-х типов



идентифицирующая

Связь называется **идентифицирующей**, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью.



неидентифицирующая

Связь называется **неидентифицирующей**, если экземпляр дочерней сущности идентифицируется не через связь с родительской сущностью.

Для идентифицирующей связи атрибуты, составляющие первичный ключ родительской сущности, будут входить в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой. Для неидентифицирующей связи атрибуты, составляющие первичный ключ родительской сущности, будут входить в состав неключевых атрибутов дочерней сущности.

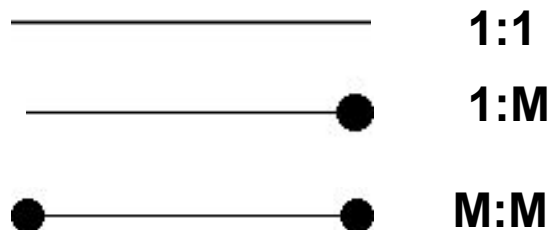
Концептуальное проектирование

Описание ER-модели данных в нотации IDEF1X (**ERWin**)

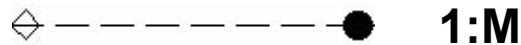
Отображение элементов ER-модели

Связи могут отображать **мощность**

Мощность связи - отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности.



Допустимость пустых (**NULL**) значений в **неидентифицирующих** связях **ERwin** изображает пустым ромбиком на дуге связи со стороны родительской сущности.

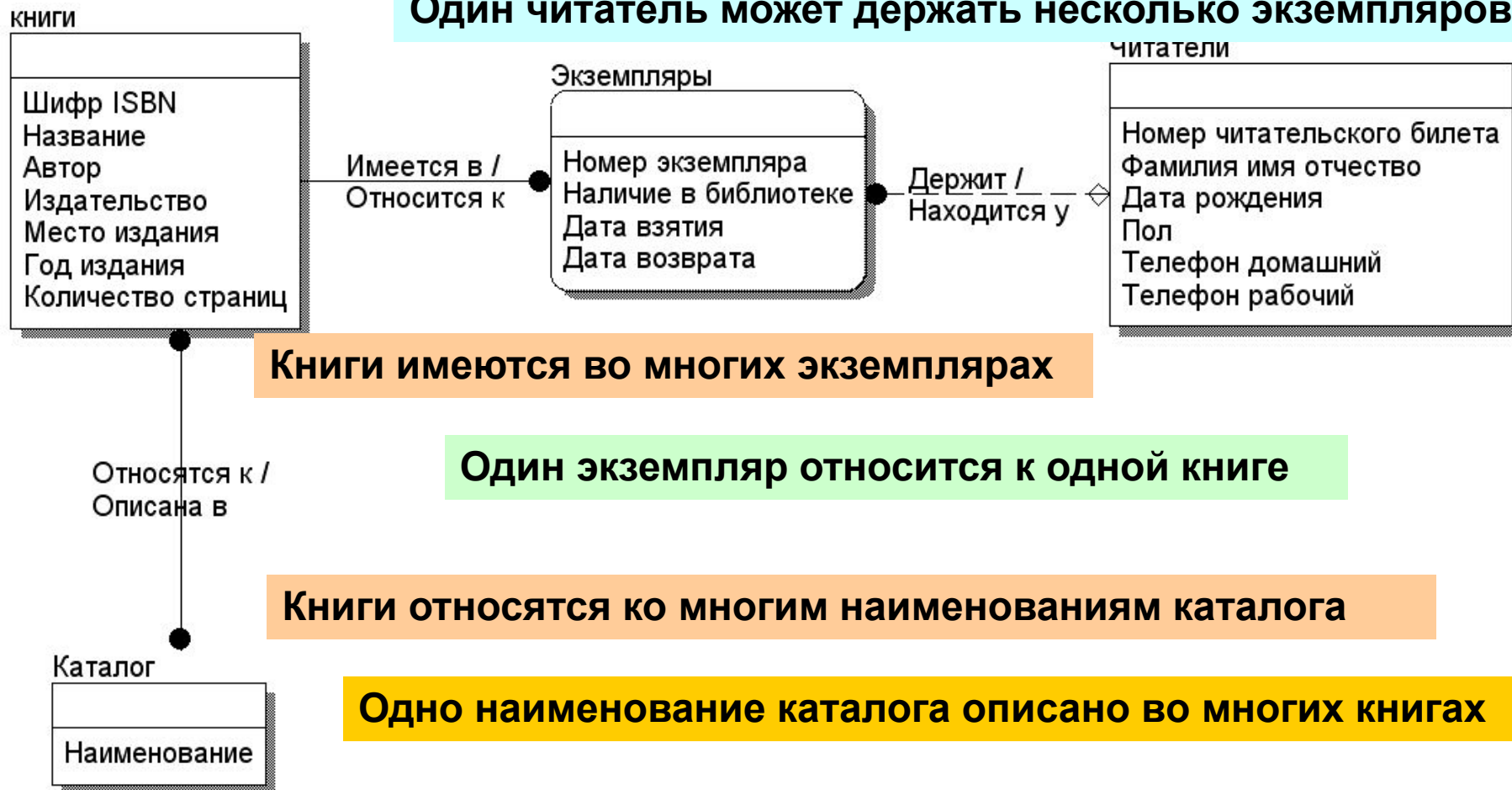


Концептуальное проектирование

Пример ER-модели данных ИС «библиотека» в нотации IDEF1X (ERWin)

Один экземпляр может находиться у одного читателя

Один читатель может держать несколько экземпляров



Концептуальное проектирование

Существует 2 подхода к реализации этого этапа проектирования

- функциональный подход
(восходящий или
снизу-вверх (bottom-up
design))

реализует принцип «от задачи» -
определения атрибутов, которые на
основании анализа группируются в
исходные отношения.

- предметный подход
(нисходящий или
сверху-вниз (top-down
design))

реализует принцип «от проблемы» -
определения объектов (или
сущностей) предметной области,
отношений между ними и
выявление атрибутов объектов.

Концептуальное проектирование

Пример функционального подхода проектирования БД ИС торговой фирмы

Выявлены группы атрибутов:

группа, характеризующая **заказ**, группа, характеризующая приход **товаров**

Дата заказа

Номер заказа

Организация Заказчик

Адрес заказчика, тел

ФИО контактного лица, должность

Должность, ФИО сотрудника

Наименование товара

Спецификация товара

Количество

Цена

Сумма

Общая сумма

Наименование товара

Спецификация товара

Организация - поставщик

Адрес поставщика, тел.

Должность, ФИО контактного лица

Дата поставки

Номер накладной

Количество прихода

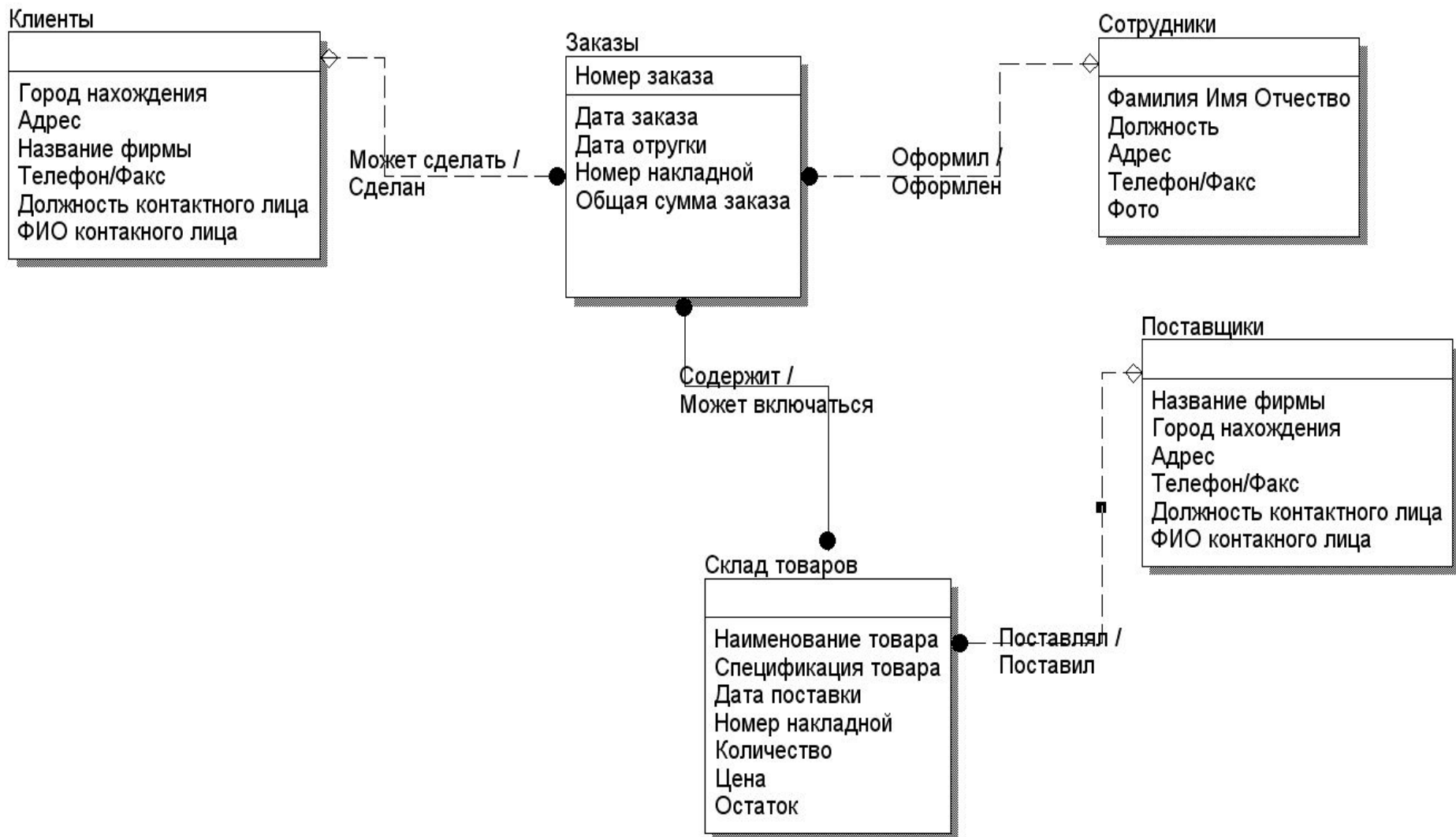
Цена

Сумма

Концептуальное проектирование

Пример предметного подхода проектирования БД ИС торговой фирмы

Выявлены объекты, их атрибуты и отношения:



Даталогическое проектирование

Основная цель этапа – разработка схемы базы данных.

Схема БД – это набор взаимосвязанных отношений, в которых определены все атрибуты, их типы и ограничения целостности, заданы первичные и вторичные ключи, определены индексы.

Создание схемы базы данных выполняется в 2 этапа.

1 этап. Этап синтеза \implies

Создание исходных отношений по результатам предыдущего этапа

2 этап. Этап декомпозиции. \implies

Замена исходных отношений другим множеством отношений с целью получения **корректной схемы БД**

Корректной схемой БД наз. схему, в которой отсутствуют нежелательные зависимости между атрибутами отношений.

Этап синтеза

Этап синтеза представляет собой процесс создания исходных схем отношений

При использовании функционального подхода это

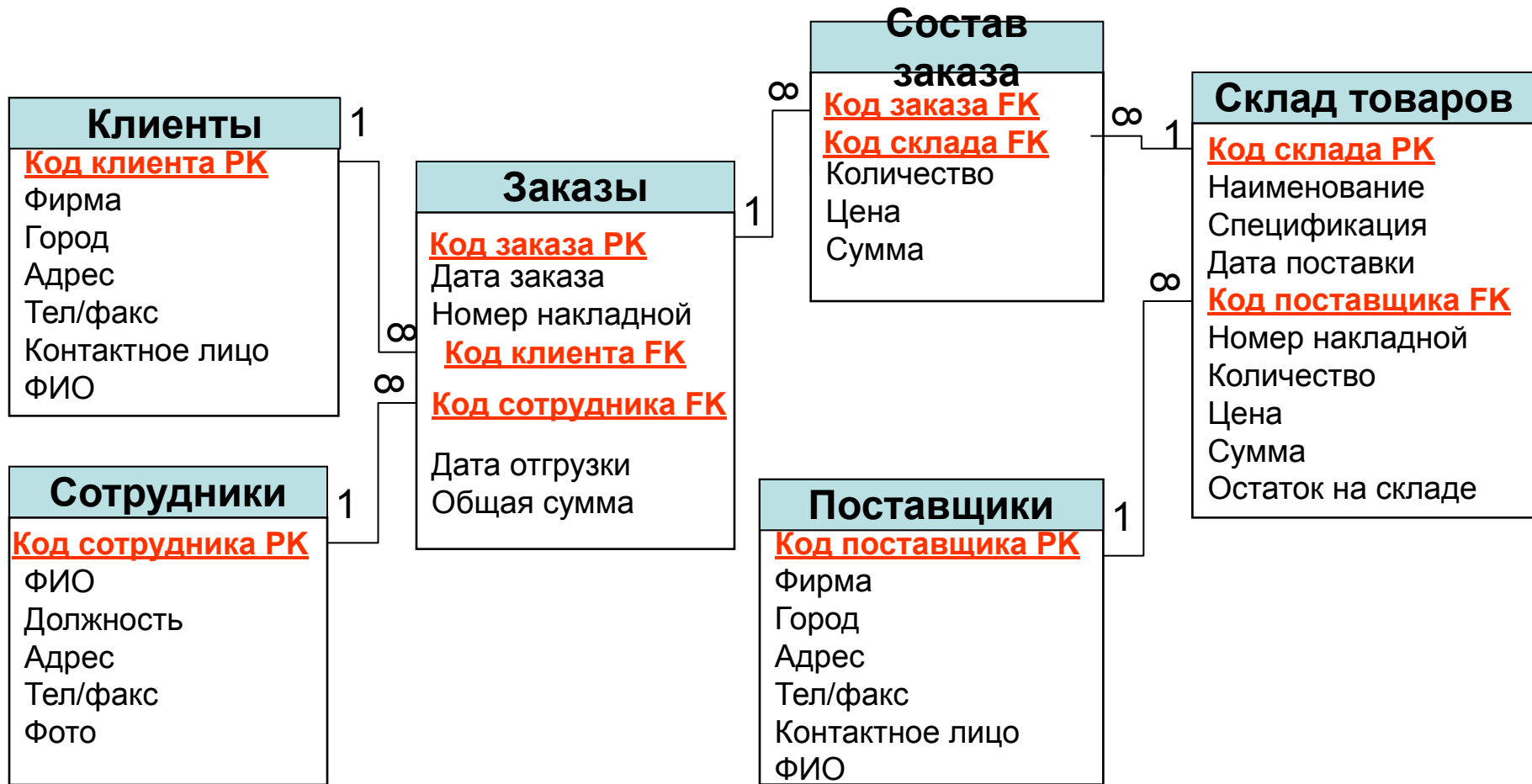
- перенос сгруппированных атрибутов в соответствующую таблицу,**
- определение первичных ключей в таблицах,**
- определение общих атрибутов, по которым устанавливаются связи.**

При использовании предметного подхода это

- замена объектов (сущностей) на таблицы,**
- определение первичных ключей в таблицах,**
- замена связей многие-ко-многим промежуточными таблицами, в которые включаются первичные атрибуты соединяемых таблиц,**

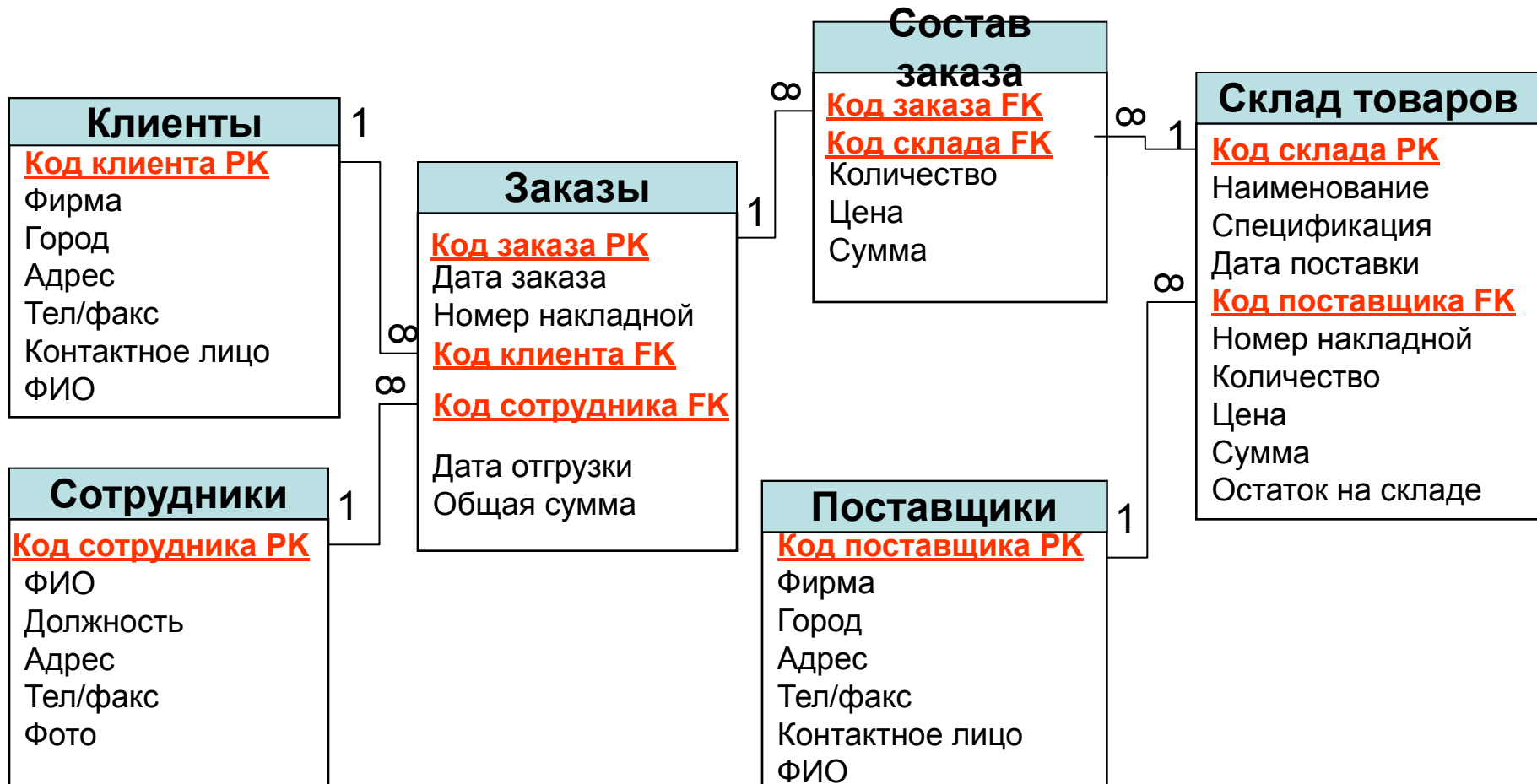
Этап синтеза

Пример предметного подхода проектирования БД ИС торговой фирмы



Этап синтеза

Пример предметного подхода проектирования БД ИС торговой фирмы



Этап декомпозиции

Этап декомпозиции представляет собой процесс последовательной нормализации схем отношений

Каждому этапу нормализации соответствует своя нормальная форма

Каждая нормальная форма обладает следующими свойствами

Каждая следующая нормальная форма улучшает в некотором смысле свойства предыдущей

При переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются

Сохраняется **эквивалентность** схемы БД при переходе к следующей нормальной форме

Схема БД называется эквивалентной, если содержание исходной схемы БД может быть получено путем естественного соединения отношений, входящих в результирующую схему, и при этом число кортежей в исходной схеме останется неизменным.

Этап декомпозиции

В теории реляционных баз данных разработана следующая последовательность нормальных форм (НФ):

- первая нормальная форма (1НФ)

- вторая нормальная форма (2НФ)

- третья нормальная форма (3НФ)

- нормальная форма Бойса-Кодда (БКНФ)

- четвертая нормальная форма (4НФ)

- пятая нормальная форма (5НФ)

Этап декомпозиции

Эквивалентные преобразования в нормальных формах основаны на анализе **функциональных зависимостей** между атрибутами отношения

Функциональная зависимость набора атрибутов B от набора атрибутов A отношения R $R.A \rightarrow R.B$ или $A \rightarrow B$

называется такое соотношение проекций $R[A]$ и $R[B]$, при котором в каждый момент времени любому элементу проекций $R[A]$ соответствует только один элемент проекций $R[B]$, входящий вместе с ним в какой-либо кортеж отношения R

Этап декомпозиции

Пусть имеется следующее отношение R с набором данных

A	B	C	D
a ₁	b ₁	c ₁	d ₁
a ₂	b ₂	c ₁	d ₁
a ₁	b ₁	c ₂	d ₂
a ₃	b ₁	c ₂	d ₃
a ₂	b ₂	c ₃	d ₂
a ₁	b ₁	c ₃	d ₄
a ₄	b ₃	c ₄	d ₂

Определим функциональные зависимости A → B и B → A.

Если каждому значению из A соответствует единственное значение из B, то A → B, если нет – то A ↗ B.

...

Функциональные зависимости определяются не на текущем состоянии БД, а на всевозможных её состояниях.

Функциональные зависимости определяются исходя из глубокого анализа предметной области.

Этап декомпозиции

Пусть имеется следующее отношение

R (Имя, Дата рождения, Знак зодиака)

Определим функциональные зависимости **Знака зодиака от Даты рождения**

решение

(Дата рождения) \twoheadrightarrow (Знак зодиака)



Функциональные зависимости определяются исходя из глубокого анализа предметной области.

Знак зодиака определяется по месяцу и дню рождения!

Этап декомпозиции

1НФ

Отношение находится в 1НФ тогда и только тогда, когда на пересечении каждого столбца и каждой строки находится только элементарные значения атрибутов.

Признаки нахождения отношения в 1НФ

1. Все поля атомарны
2. Отсутствуют повторяющиеся группы
3. Определён первичный ключ
4. Все атрибуты зависят от первичного ключа

Этап декомпозиции

1НФ

Например, пусть имеется таблица расписания

Преподаватель	День недели	Номер пары	Дисциплина	Тип занятий	Группа
Петров	пн,вт,ср	1, 1, 2	ПА, АВМ, ОПБД	лк, лб, лб	8032, 7032, 7033
Сидоров	вт,вт,ср	2, 3, 1	АВМ,ПА, ОПБД	лк, лб, лб	7033, 7032, 8032
Иванов	пн,ср,пт	2, 3, 1	АОС, ПА, АОС	лк, лб, лб	7032, 8032, 7033

Этап декомпозиции

1НФ

Приведение таблицы расписания к 1НФ

Преподаватель РК	День недели РК	Номер пары РК	Дисциплина	Тип занятий	Группа
Петров	пн	1	ПА	лк	8032
Петров	вт	1	АВМ	лб	7032
Петров	ср	2	ОПБД	лб	7033
Сидоров	вт	2	АВМ	лк	7033
Сидоров	вт	3	ПА	лб	7032
Сидоров	ср	1	ОПБД	лб	8032
Иванов	пн	2	АОС	лк	7032
Иванов	ср	3	ПА	лб	8032
Иванов	пт	1	АОС	лб	7033

Этап декомпозиции

2НФ

Отношение находится в 2НФ тогда и только тогда, когда оно находится в 1НФ и не содержит неполных функциональных зависимостей первичных атрибутов от атрибутов первичного ключа.

Полная функциональная зависимость – это когда значение в каждом неключевом столбце однозначно определяется значением всех столбцов первичного ключа

Пример. Отношение R моделирующее сдачу сессии со следующими атрибутами

R(ФИО; $\underbrace{\text{Ном.ЗК}}_{\text{РК}}; \text{Группа}; \underbrace{\text{Дисциплина}}_{\text{РК}}; \text{Оценка})$

$\text{НомЗК} \rightarrow \text{ФИО}$

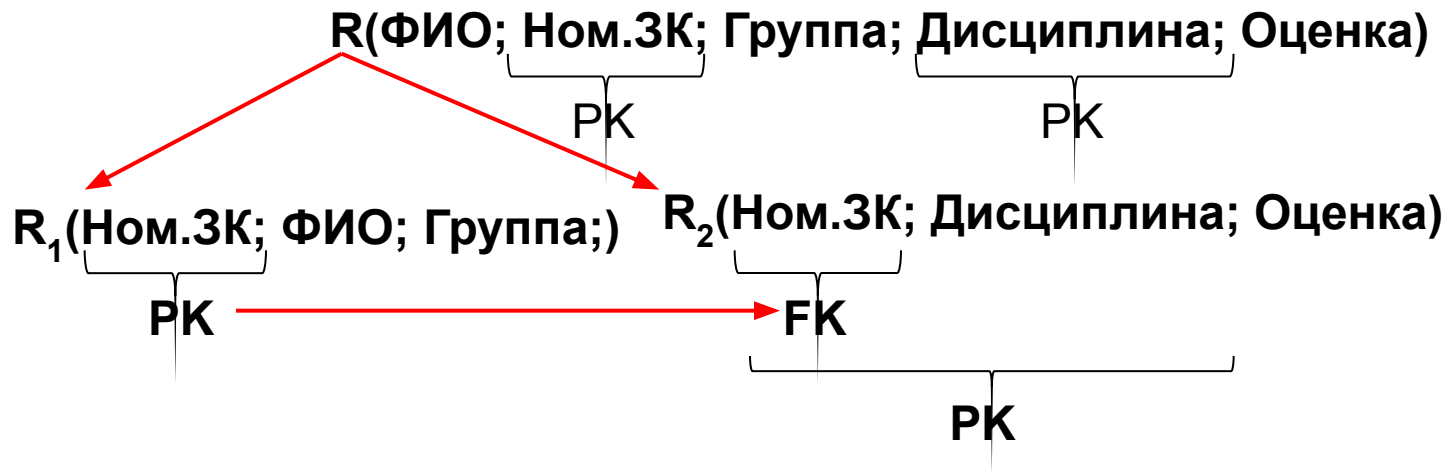
$\text{НомЗК} \rightarrow \text{Группа}$

Для приведения отношения ко 2НФ следует разбить его на проекции: переместить неключевые атрибуты, между которыми существует неполная зависимость, в другое отношение

Этап декомпозиции

2НФ

Пример. Отношение R моделирующее сдачу сессии со следующими атрибутами



Этап декомпозиции

3НФ

Отношение находится в 3НФ тогда и только тогда, когда оно находится в 2НФ и не содержит транзитивных зависимостей (ни один неключевой атрибут не зависит от другого неключевого атрибута, а зависит только от первичного ключа).

Функциональная зависимость $A \rightarrow B$ называется транзитивной, если существует набор атрибутов C такой, что C не является подмножеством A и не включает в себя B $C \not\subseteq A$, $B \not\subseteq C$,
существует зависимости $A \rightarrow C$ и $C \rightarrow B$ и
не существует зависимость $C \rightarrow A$

Этап декомпозиции

Пример. Пусть имеется отношение R

R(ФИО; $\underbrace{\text{Ном.ЗК}; \text{Специальность}; \text{Группа}}$)

PK

НомЗК → *ФИО*

НомЗК → *Группа*

НомЗК → *Специальность*

Группа → *Специальность*

НомЗК → *Группа* → *Специальность*

Для приведения отношения ко 3НФ следует разбить его на проекции: переместить неключевые атрибуты, между которыми существует зависимость, в другое отношение

Этап декомпозиции

3НФ



Этап декомпозиции

БКНФ

Отношение находится в БКНФ тогда и только тогда, когда оно находится в 3НФ и каждый детерминант отношения является возможным ключом отношения

Детерминантом наз. любой атрибут, от значения которого зависят значения других атрибутов

Условия, когда отношение находится в 3НФ, но не находится в БКНФ:

- 1. Отношение имеет 2 или более потенциальных ключа;**
- 2. Потенциальные ключи являются составными.**
- 3. Потенциальные ключи перекрываются, т.е. имеют, по крайней мере, один общий атрибут.**

Этап декомпозиции

Пример. Пусть имеется отношение R, моделирующее сдачу экзаменационной сессии со следующими условиями:

- можно сдавать экзамен по одному предмету несколько раз
- для идентификации студента используется уникальный номер

R(ИД; Ном.ЗК; Дисциплина; Дата; Оценка)

PK

PK

PK

Потенциальные ключи:

Ном.ЗК+ Дисциплина + Дата

ИД+ Дисциплина + Дата

Детерминанты не являющиеся потенциальными ключами

Функциональные зависимости

$\text{НомЗК} + \text{Дисциплина} + \text{Дата} \rightarrow \text{Оценка}$

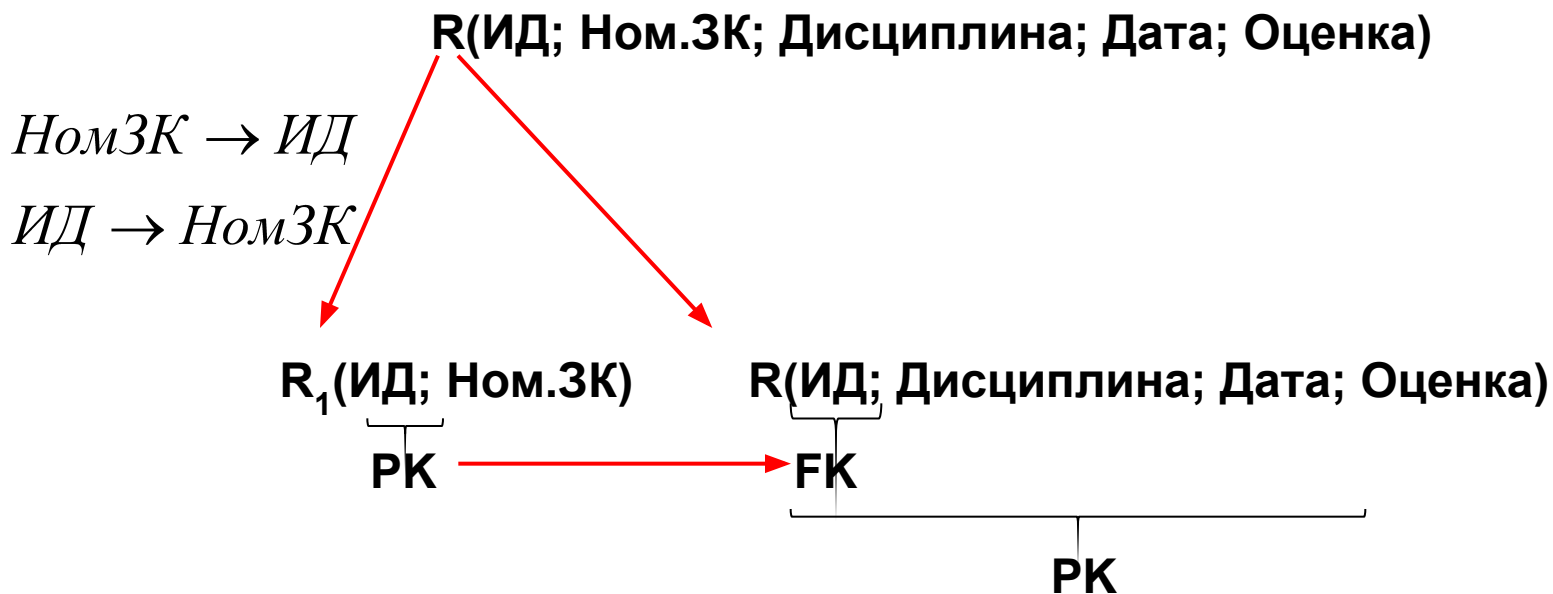
$\text{ИД} + \text{Дисциплина} + \text{Дата} \rightarrow \text{Оценка}$

$\text{НомЗК} \rightarrow \text{ИД}$

$\text{ИД} \rightarrow \text{НомЗК}$

Этап декомпозиции

Для приведения отношения ко БКНФ следует разбить его на проекции: переместить в другое отношение зависимую часть с детерминантом, который не является потенциальным ключом



Этап декомпозиции

4НФ

Отношение находится в 4НФ тогда и только тогда, когда оно находится в БКНФ и если в случае существования многозначной зависимости $A \twoheadrightarrow B$ все остальные атрибуты функционально зависят от A (т.е. если существует многозначная зависимость, то только одного атрибута).

В отношении $R(A,B,C)$ существует многозначная зависимость B от A ($A \twoheadrightarrow B$) в том и только в том случае, если множество значений B , соответствующее паре значений A и C , зависит только от A и не зависит от C

Этап декомпозиции

4НФ

Пример. Пусть имеется отношение R

R(Ном.ЗК; Группа; Дисциплина)



Здесь имеются многозначные зависимости

Группа ->> Дисциплина

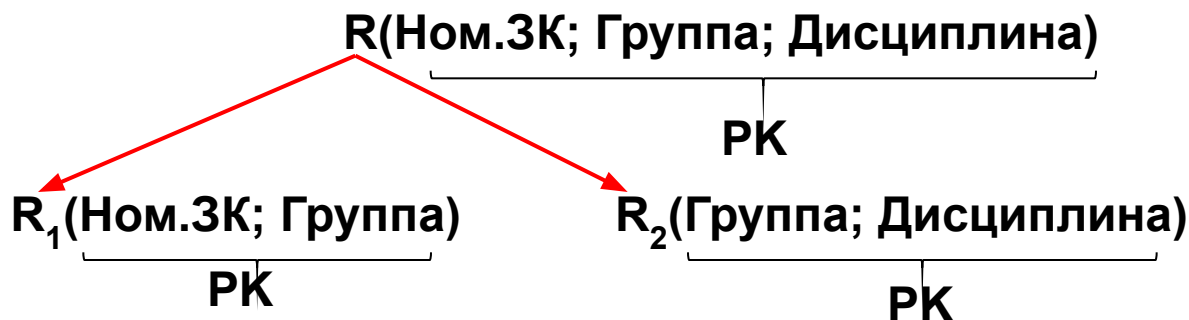
Группа ->> Ном.ЗК

Ном.ЗК	Группа	Дисциплина
20-01	20	АОС
20-02	20	АОС
20-03	20	АОС
20-01	20	АВМ
20-02	20	АВМ
20-03	20	АВМ
20-01	20	ОПБД
20-02	20	ОПБД
20-03	20	ОПБД
21-01	21	АОС
21-02	21	АОС
21-01	21	АВМ
21-02	21	АВМ
21-01	21	ОПБД
21-02	21	ОПБД

Этап декомпозиции

4НФ

Теорема Фейджина: Отношение $R(A,B,C)$ можно спроецировать без потерь в отношения $R_1(A,B)$ и $R_2(A,C)$ в том и только том случае, когда существует многозначная зависимость $A \twoheadrightarrow B$ и $A \twoheadrightarrow C$



Ном.3К	Группа
20-01	20
20-02	20
20-03	20
21-01	21
21-02	21

Группа	Дисциплина
20	АОС
20	АВМ
20	ОПБД
21	АВМ
21	ОПБД

Этап декомпозиции

Пример нормализации таблиц БД ИС торговой фирмы

Заказы
Код заказа РК
Дата заказа
Номер накладной
Фирма заказчика
Город
Адрес заказчика
Тел-факс
Контактное лицо (должность)
ФИО контактного лица
ФИО сотрудника
Должность сотрудника
Наименование товара
Спецификация товара
Количество
Цена
Сумма
Общая сумма

К 1НФ →

Заказы
Код заказа РК
Дата заказа
Номер накладной
Фирма заказчика
Город
Адрес заказчика
Тел-факс
Контактное лицо (должность)
ФИО контактного лица
ФИО сотрудника
Должность сотрудника
Общая сумма

1

∞

Состав заказа	
Код заказа FK	PK
Наименование товара	PK
Спецификация товара	PK
Количество	
Цена	
Сумма	

Повторяющаяся группа атрибутов

Этап декомпозиции

Пример нормализации таблиц БД ИС торговой фирмы

Заказы
Код заказа РК
Дата заказа
Номер накладной
Фирма заказчика
Город
Адрес заказчика
Тел-факс
Контактное лицо (должность)
ФИО контактного лица
ФИО сотрудника
Должность сотрудника
Общая сумма

К 3НФ →

Заказы
Код заказа РК
Дата заказа
Номер накладной
Код клиента FK
Код сотрудника FK
Общая сумма



Транзитивные зависимости:

Код заказа -> Фирма-> Город

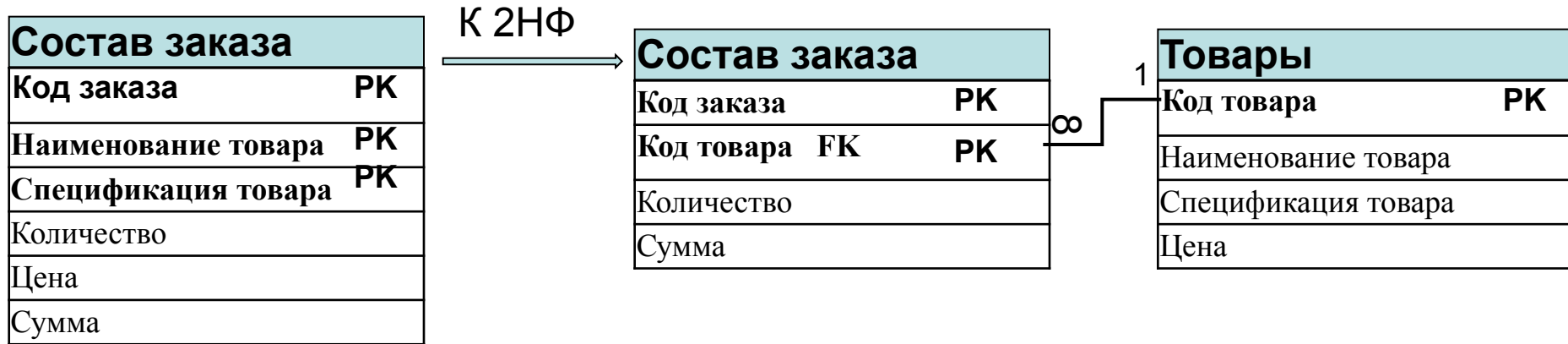
Код заказа -> Фирма-> Адрес

...

Код заказа -> ФИО сотрудника -> Должность сотрудника

Этап декомпозиции

Пример нормализации таблиц БД ИС торговой фирмы



Полные функциональные зависимости:

(Код заказа, Наименование товара, Спецификация товара --> Количество, Сумма)

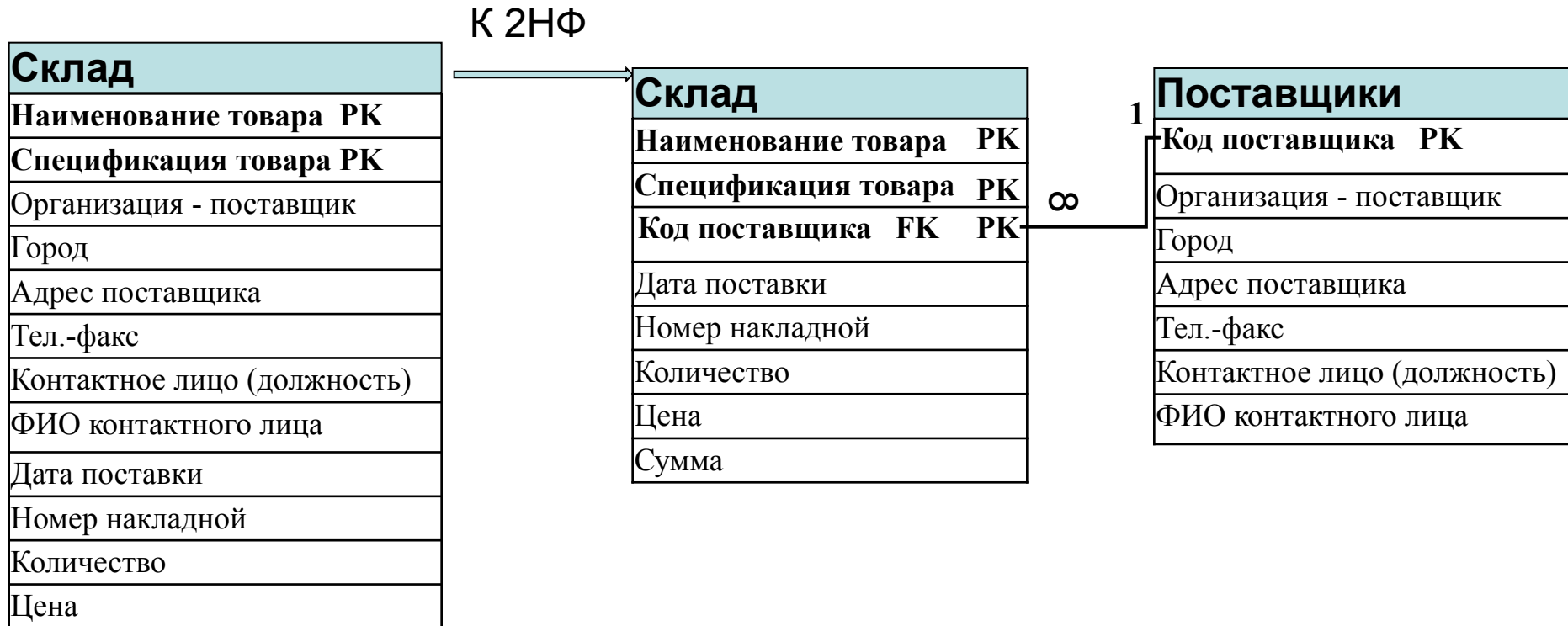
Не полные функциональные зависимости:

Наименование товара, Спецификация товара ->Цена

(Код заказа-/-> Цена)

Этап декомпозиции

Пример нормализации таблиц БД ИС торговой фирмы



Транзитивные зависимости:

Наименование товара, Спецификация товара -> Организация-> Город

...

Этап декомпозиции

Пример нормализации таблиц БД ИС торговой фирмы

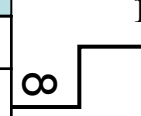
К 3НФ

Склад	
Код товара	PK
Наименование товара	
Спецификация товара	
Код поставщика	
Дата поставки	
Номер накладной	
Количество	
Цена	
Сумма	



Склад	
Код склада	PK
Код товара	FK
Спецификация товара	
Код поставщика	
Дата поставки	
Номер накладной	
Количество	
Цена	
Сумма	
Остаток на складе	

Каталог товаров	
Код товара	PK
Наименование товара	



Транзитивные зависимости:

Этап декомпозиции

Конечная схема БД после этапа нормализации

