

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Выбор СУБД - критерии

1. Моделирование данных (используемая модель данных, возможность реализации хранимых процедур и триггеров, предусмотренные типы данных, реализация языка запросов)
2. Особенности архитектуры и функциональные возможности (мобильность, масштабируемость, распределенность)
3. Контроль работы системы (использования памяти)
4. Особенности разработки приложений (встроенные или сопутствующие средства разработки приложений, средства проектирования, многоязыковая поддержка, возможность разработки веб-приложений, поддерживаемые языки программирования)

Выбор СУБД - критерии

1. Производительность (рейтинг ТРС – отношение запросов обрабатываемых за единицу времени к стоимости, возможность параллельной архитектуры, возможность оптимизации запросов)
2. Надежность (восстановление после сбоев, резервное копирование, откат изменений, многоуровневая система защиты)
3. Требования к рабочей среде (поддерживаемые аппаратные платформы, минимальные требования к оборудованию, максимальный размер адресуемой памяти, операционные системы под которыми может работать СУБД)
4. Другие критерии (полнота и качество документации, локализованность, стоимость, распространённость, производитель)

Логическое (дatalogическое) проектирование

Нормализацией отношений информационной модели предметной области – это процесс превращения объектов инфологической модели (сущности и связи) в логические таблицы базы данных.

Требования к логическому проектированию:

1. Первичные ключи отношений должны быть минимальными (требование минимальности первичных ключей).
2. Число отношений базы данных должно по возможности давать наименьшую избыточность данных (требование надежности данных).
3. Число отношений базы данных не должно приводить к потере производительности системы (требование производительности системы).

Требования к логическому проектированию:

3. Данные не должны быть противоречивыми, т.е. при выполнении *операций включения, удаления и обновления данных* их потенциальная противоречивость должна быть сведена к минимуму (требования непротиворечивости данных).
4. Схема отношений базы данных должна быть устойчивой, способной адаптироваться к изменениям при ее расширении дополнительными атрибутами (требование гибкости структуры базы данных).
5. Разброс времени реакции на различные запросы к базе данных не должен быть большим (требование производительности системы).
6. Данные должны правильно отражать состояние предметной области базы данных в каждый конкретный момент времени (требование актуальности данных).

Аномалии

Код_клиента	Фамилия	Адрес	Код_товара	Название	Количество
1	Иванов	Ленина, 40	1	Телевизор	1
2	Сидоров	Советская, 39	1	Телевизор	3
1	Иванов	Ленина, 40	2	Утюг	3
2	Сидоров	Советская, 39	2	Утюг	2
3	Петров	Лесная, 5	1	Телевизор	6

Аномалии удаления: при удалении информации о всех покупателях, купивших определенный товар, теряется информация об этом

товаре

Аномалии обновления: при изменении адреса покупателя нужно вносить изменения во все строки с этим покупателем

Аномалии добавления: нельзя добавить информацию о товаре, если его никто не заказал.

Первая нормальная форма

Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения являются простыми (требование атомарности атрибутов в реляционной модели), т.е. не имеют компонентов.

Иными словами, домен атрибута должен состоять из неделимых значений и не может включать в себя множество значений из более элементарных доменов.

Пример

Код_заказа	Дата	ФИО_клиента	Позиция заказа	
			наименование	количество
1	20.04.2014	Иванов Иван Иванович	карандаш	10
			ластик	15
2	21.05.2014	Тарасова Ольга Васильевна	пенал	1
			карандаш	20
			линейка	25

Приведение к 1НФ

Код_заказа	Дата	ФИО_клиента	Наименование_товара	Количество_товара
1	20.04.2014	Иванов Иван Иванович	карандаш	10
1	20.04.2014	Иванов Иван Иванович	ластик	15
2	21.05.2014	Тарасова Ольга Васильевна	пенал	1
2	21.05.2014	Тарасова Ольга Васильевна	карандаш	20
2	21.05.2014	Тарасова Ольга Васильевна	линейка	25

Функциональная зависимость

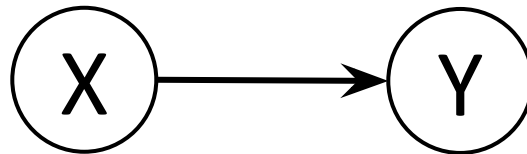
Отношение задается списком имен атрибутов – *схемой отношения*.

Функциональной зависимости данных – это ограничения на возможные взаимосвязи между данными, которые могут быть текущими значениями *схемы отношений*.

Определение ФЗ

Пусть $r(A, B, C, D, \dots)$ – схема отношения R ,
а X и Y – подмножества r . Говорят,
что X функционально определяет Y , если каждому
значению атрибутов кортежа
отношения из X соответствует не более одного
значения атрибутов того же кортежа
отношения из Y . Такая ФЗ обозначается как

$$F: X \rightarrow Y$$



Пример ФЗ

ТОВАРЫ_НА_СКЛАДАХ(Код_товара, Название, Цена, Склад, Адрес, Количество)

Известно, что:

- Каждому коду товара соответствует определенное название и цена товара
- У каждого склада есть свой адрес
- Каждому товару на каждом складе соответствует определенное количество товара

Пример ФЗ

<u>Код_товар</u>	Название	Цена	<u>Склад</u>	Адрес	Количество
1	Телевизор	20000	1	Ленина,13	34
1	Телевизор	20000	2	Гагарина,85	33
1	Телевизор	20000	3	Советская, 44	23
2	Пылесос	7000	1	Ленина,13	1
2	Пылесос	7000	3	Советская, 44	12
3	Ноутбук	30000	2	Гагарина,85	24
3	Ноутбук	30000	3	Советская, 44	22

{Цена, Название} функционально зависит от Код_товара:

Код_товара \rightarrow {Цена, Название}

Адрес функционально зависит от Склад: Склад \rightarrow Адрес

Количество функционально зависит от {Код_товара,

Склад}: {Код_товара, Склад} \rightarrow Количество

Частные и полные функциональные зависимости

Полная ФЗ: неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Частичная ФЗ: неключевой атрибут зависит от части составного ключа, то говорят о

Примеры полной и частной ФЗ

ТОВАРЫ_НА_СКЛАДАХ(Код_товара,Склад, Название, Цена, Адрес, Количество)

<u>Код_товар</u>	Название	Цена	<u>Склад</u>	Адрес	Количество
1	Телевизор	20000	1	Ленина,13	34
1	Телевизор	20000	2	Гагарина,85	33
1	Телевизор	20000	3	Советская, 44	23
2	Пылесос	7000	1	Ленина,13	1
2	Пылесос	7000	3	Советская, 44	12
3	Ноутбук	30000	2	Гагарина,85	24
3	Ноутбук	30000	3	Советская, 44	22

Полная ФЗ:

{Код_товара, Склад} → Количество

Частичные ФЗ:

Код_товара → {Цена, Название}

Склад → Адрес

Вторая нормальная форма

Отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ, и все неключевые атрибуты отношения функционально полностью зависят от составного ключа отношения.

Иными словами, 2НФ требует, чтобы отношение не содержало частичных ФЗ.

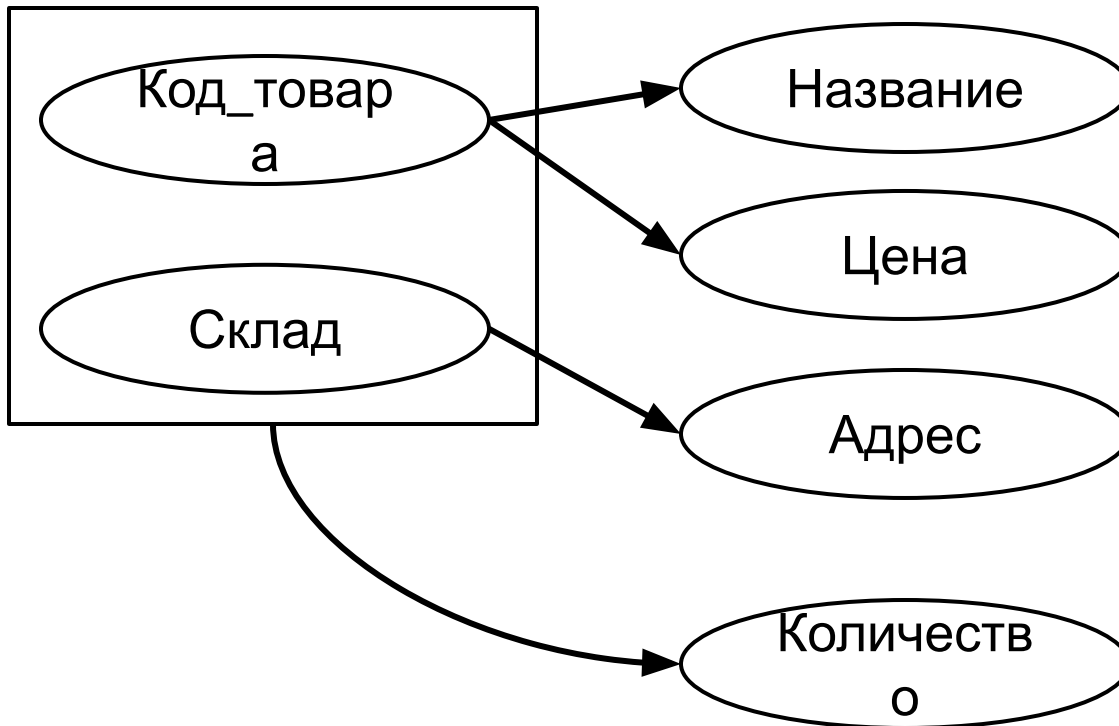
Декомпозиция

Декомпозицией схемы отношения

$R = \{A, B, C, D, \dots\}$ называется замена ее совокупностью подмножеств R , таких, что их объединение дает R . При этом допускается, чтобы подмножества были пересекающимися.

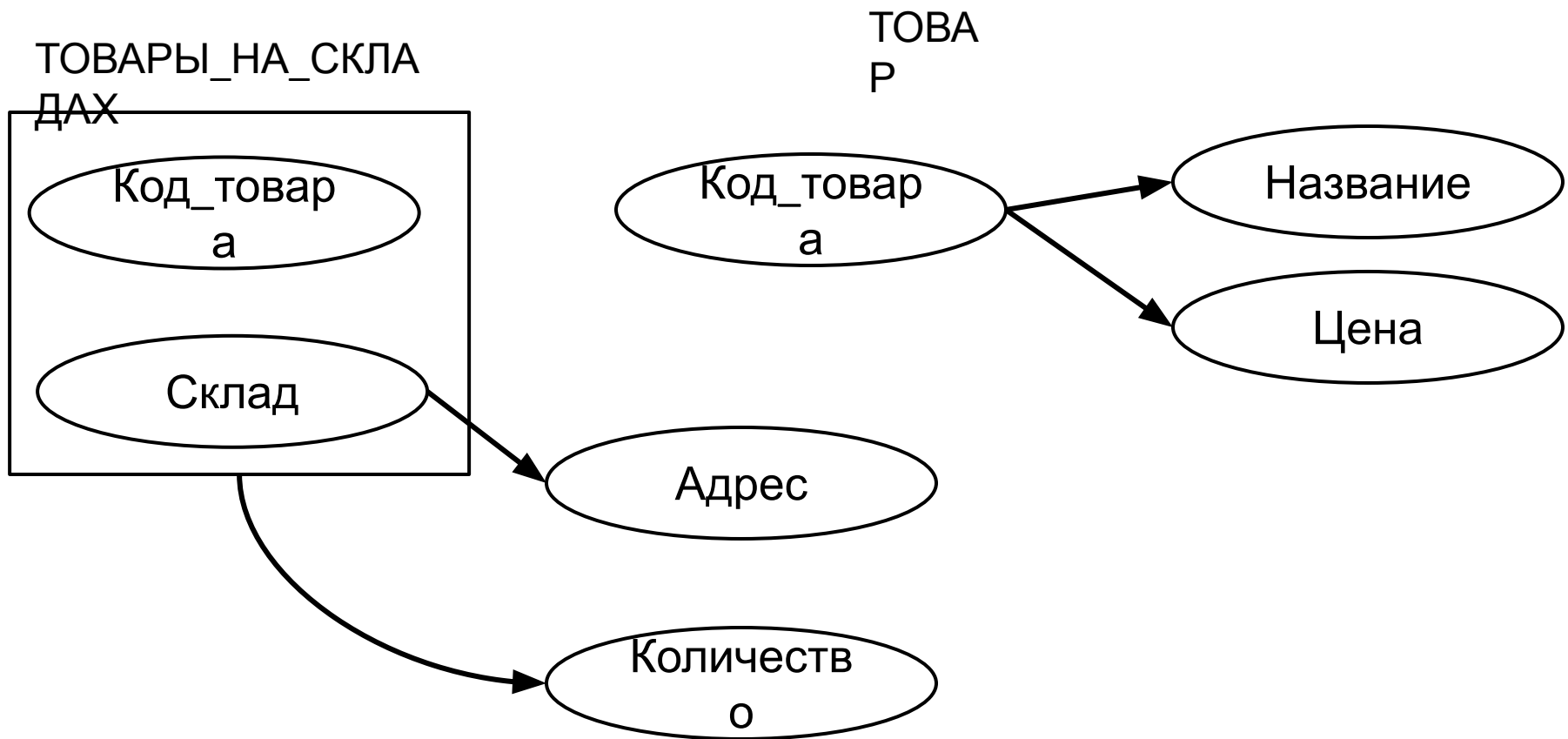
Пример приведения во 2НФ

ТОВАРЫ_НА_СКЛАДАХ(Код_товара, Склад, Название, Цена, Адрес, Количество)



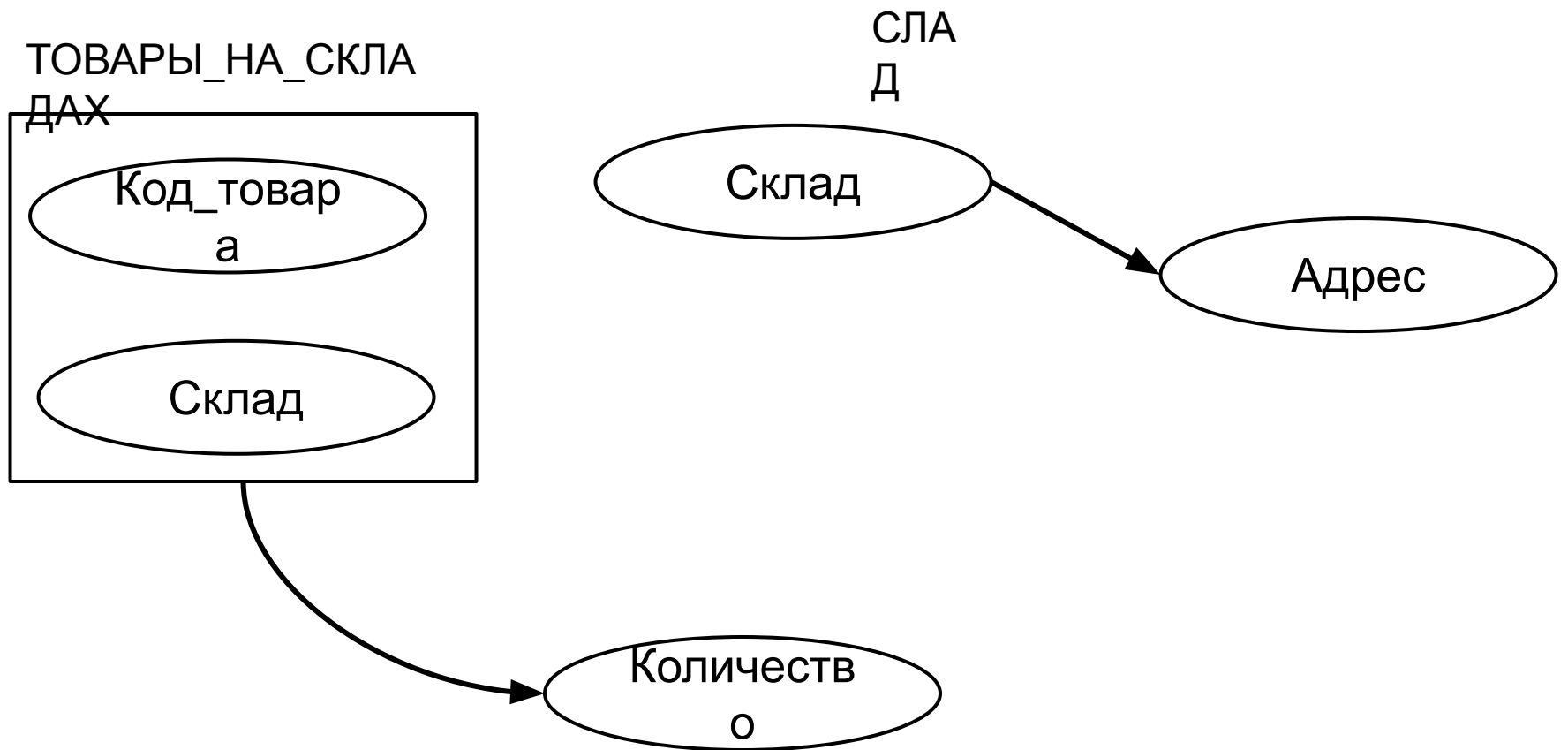
Пример приведения во 2НФ

Декомпозиция:



Пример приведения во 2НФ

Декомпозиция:

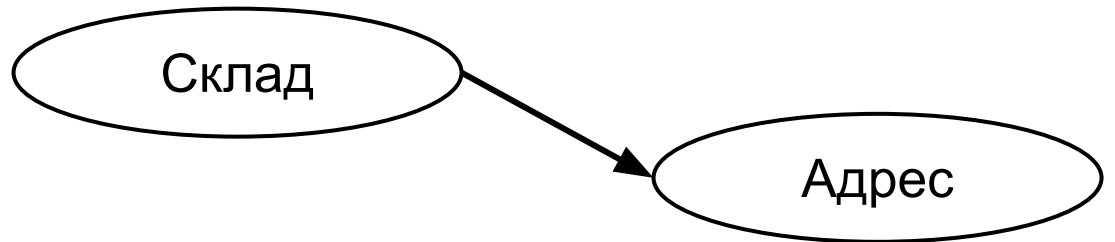


Отношения во 2НФ

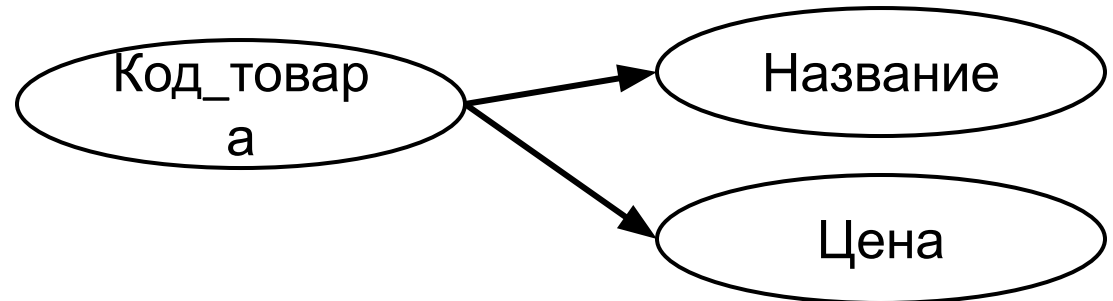
ТОВАРЫ_НА_СКЛАДАХ



СЛА
Д



ТОВА
Р



Третья нормальная форма

Отношение находится в третьей нормальной форме (3НФ), если оно находится во 2НФ, и все неключевые атрибуты отношения зависят только от первичного ключа.

Иными словами, 3НФ требует, чтобы отношение не содержало транзитивных ФЗ неключевых атрибутов от ключа.

Транзитивные зависимости

Пусть X , Y , Z – атрибуты отношения R . Если при этом имеются ФЗ $X \rightarrow Y$ и $Y \rightarrow Z$, но отсутствуют ФЗ $Z \rightarrow Y$ и $Y \rightarrow X$, то говорят, что Z *транзитивно* зависит от X .

Такие ФЗ называются *транзитивными* (Т-зависимостями).

Пример транзитивной зависимости

ТОВАР (Код товара, Название, Цена, Категория, Наценка)

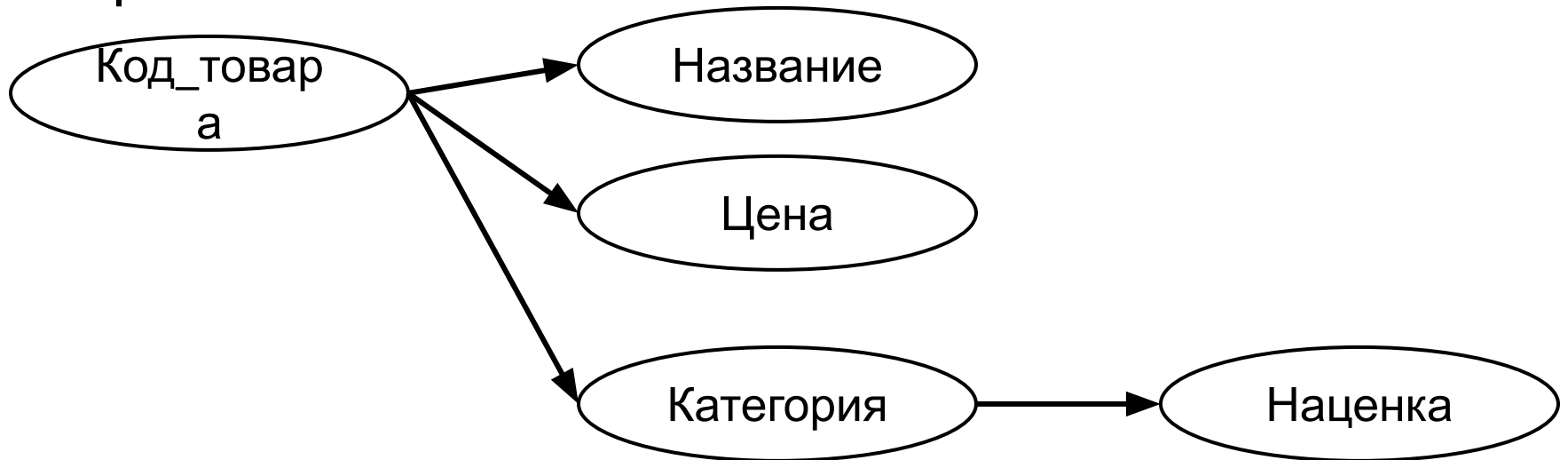
<u>Код товара</u>	Название	Цена	Категория	Наценка
1	Телевизор	20000	Телевизоры	40
2	3D телевизор	30000	Телевизоры	40
3	Миксер	4000	Техника для дома	20
4	Блендер	3000	Техника для дома	20
5	Фотоаппарат	18000	Фото и видео	50
6	Видеокамера	25000	Фото и видео	50
7	Фотопринтер	10000	Фото и видео	50

Транзитивная зависимость:

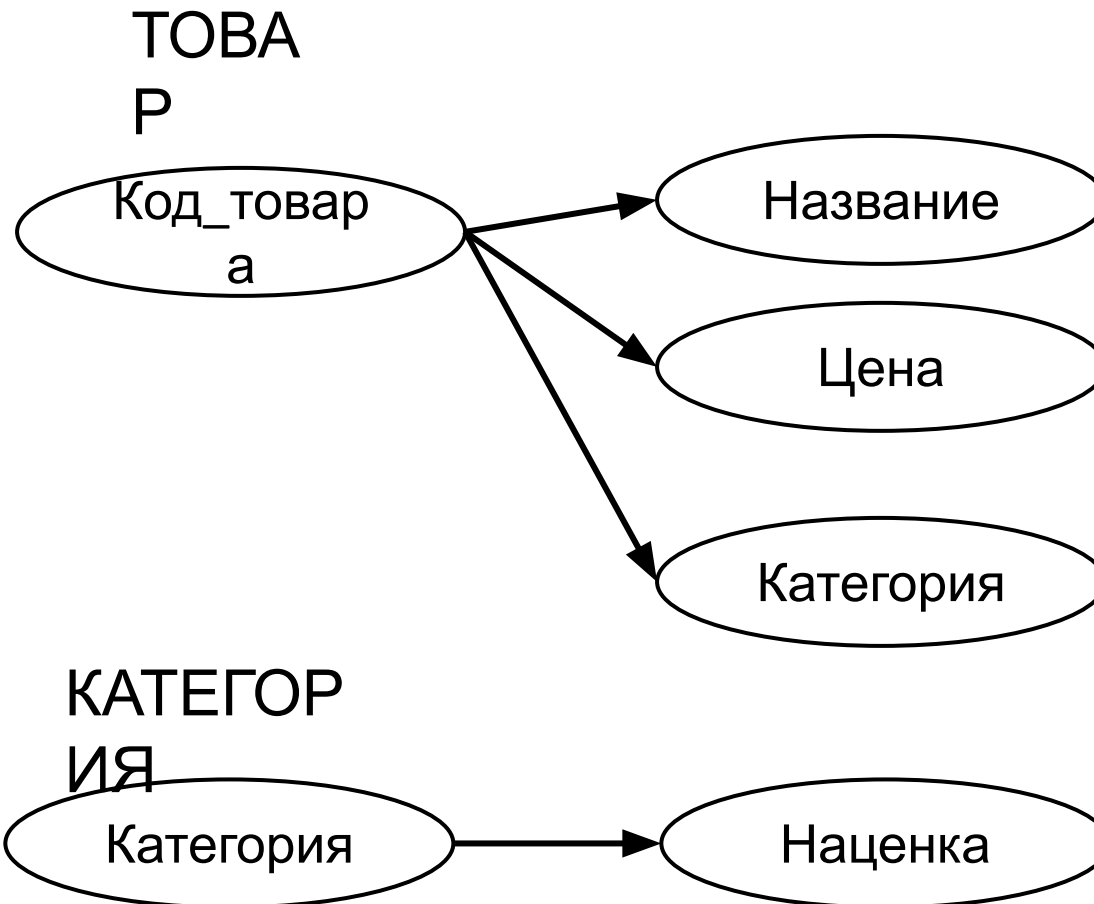
Категория → Наценка

Пример приведения к 3НФ

ТОВА
Р



Пример приведения к 3НФ



Нормальная форма Бойса-Кодда

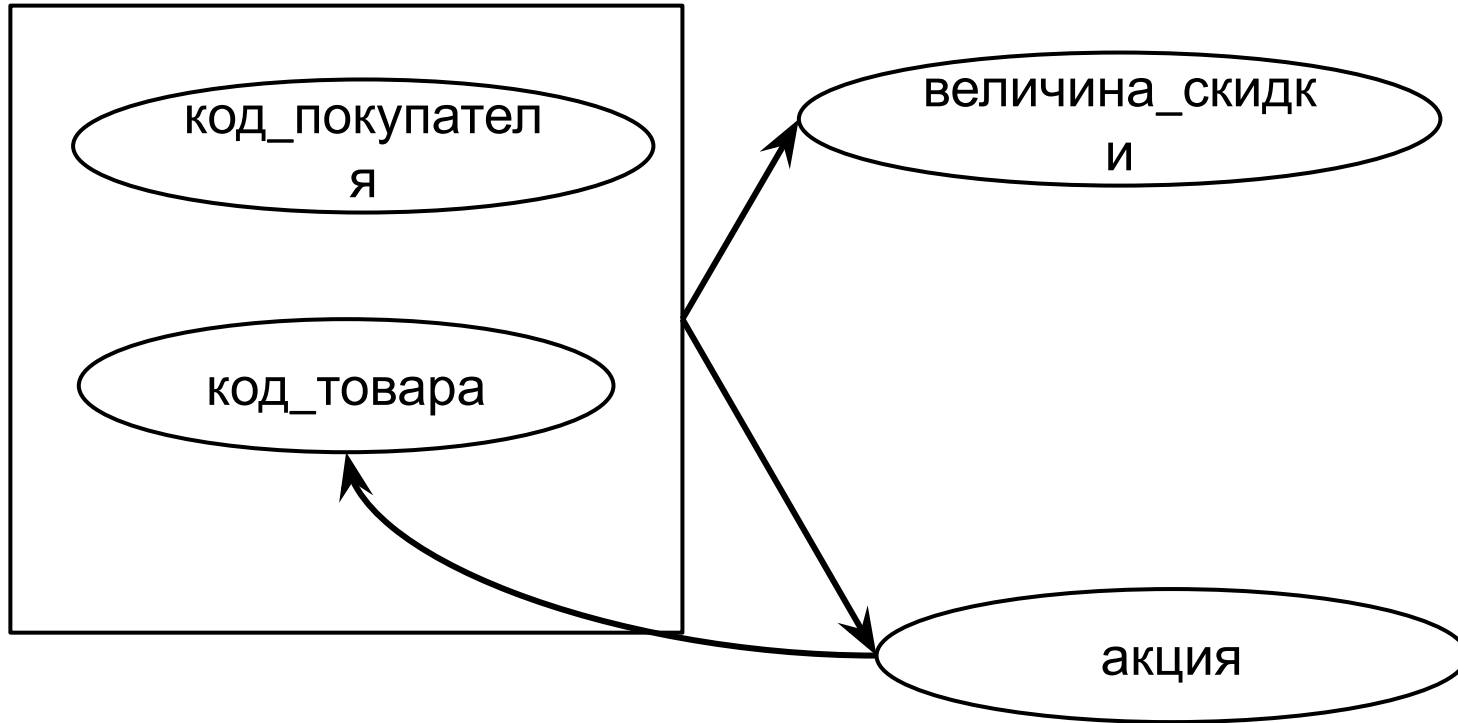
Отношение находится в нормальной форме Бойса-Кодда (НФБК), если оно находится в 3НФ, и в нем отсутствовали зависимости ключевых атрибутов от неключевых атрибутов.

Все детерминанты отношения должны быть его потенциальными ключами

Детерминант

Детерминант функциональной зависимости – это минимальная группа атрибутов, от которой зависит некоторый другой атрибут или группа атрибутов, причем эта зависимость нетривиальная.

Пример приведения в НФБК



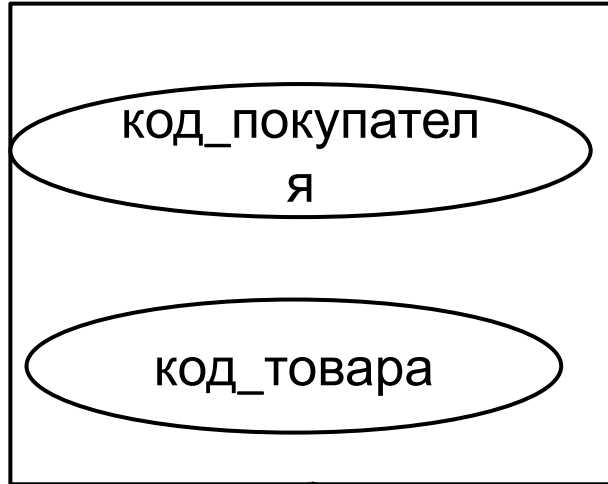
Возможные ключи:

код_покупателя, код_товара

Детерминанты:

код_покупателя код_товара

Пример приведения в НФБК

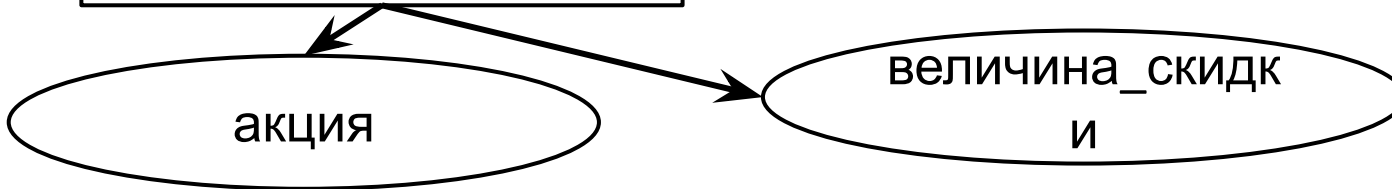


Возможные ключи:

код_покупателя, код_товара

Детерминанты:

код_покупателя, код_товара

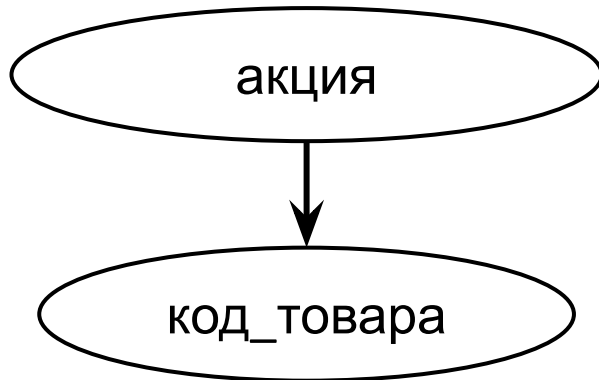


Возможные ключи:

акция

Детерминанты:

акция



Четвертая нормальная форма

Отношение находится в четвертой нормальной форме (4НФ), если оно находится в 3НФ или НФБК и все независимые многозначные ФЗ разнесены в отдельные отношения с одним и тем же ключом.

Иными словами, 4НФ применяется при наличии в отношении более чем одной многозначной ФЗ и требует, чтобы *отношение* не содержало независимых многозначных ФЗ.

Многозначные ФЗ

Пусть r – некоторая *схема отношения*, X и Y – подмножества атрибутов r . Если при заданных значениях атрибутов из $\{X\}$ существует некоторое множество, состоящее из нуля или более взаимосвязанных значений атрибутов из $\{Y\}$, никак не связанных со значениями других атрибутов этого отношения $r - X - Y$, то говорят о существовании *многозначной зависимости* между атрибутами X и Y : $MV: X \twoheadrightarrow Y$ (класс MV -зависимостей).

Пример многозначной ФЗ

МАГАЗИН_ТОВАР_СКЛАД(код_магазина, код_товара, код_склада)

Код_магазина	Код_товара	Код_склада
1	1	2
1	2	2
2	1	1
2	3	2
2	3	3
3	1	2
3	1	1

Многозначная ФЗ:

Код_магазина →→ Код_товара

Код_магазина →→ Код_склада

Приведение в 4НФ

МАГАЗИН_ТОВ

АР

Код_магазин Код_товара

а

1 1

1 2

2 1

2 3

3 1

МАГАЗИН_СКЛ

АД

Код_магазин Код_склада

а

1 2

2 1

2 2

2 3

3 2

3 1

Пятая нормальная форма

Отношение находится в *пятой нормальной форме (5НФ)*, если оно находится в 4НФ и удовлетворяет зависимости по соединению относительно своих проекций.

5НФ называют также нормальной формой с проецированием соединений.

Она используется для разрешения трех и более отношений, которые связаны более чем тремя ФЗ по типу "*многие-ко-многим*".

Зависимость соединения

Отношение $R(X, Y, \dots, Z)$ удовлетворяет зависимости соединения $*$ (X, Y, \dots, Z) в том и только в том случае, когда R восстанавливается без потерь путем соединения своих проекций на X, Y, \dots, Z .

Пример

ТОВАР_МАГАЗИН_ПОСТАВЩИК

Код_магазина	Код_товара	Код_поставщика
1	1	2
1	2	1
2	1	1
1	1	1

Проекции

МАГАЗИН_ТОВАР МАГАЗИН_ПОСТАВЩИК

ТОВАР_ПОСТАВЩИК

Код_магазина	Код_товара	Код_магазина	Код_поставщика	Код_товара	Код_поставщика
1	1	1	2	1	2
1	2	2	1	2	1
2	1	1	1	1	1

Соединение проекций

МАГАЗИН_ТОВАР и МАГАЗИН_ПОСТАВЩИК

Код_магазина	Код_товара	Код_поставщика
1	1	2
1	1	1
1	2	2
1	2	1
2	1	1

Итог

ТОВАР_МАГАЗИН_ПОСТАВЩИК

Код_магазина	Код_товара	Код_поставщика
1	1	2
1	2	1
2	1	1
1	1	1

МАГАЗИН_ТОВАР МАГАЗИН_ПОСТАВЩИК ТОВАР_ПОСТАВЩИК

Код_магазин ина	Код_товара	Код_магазин ина	Код_поставщи ка	Код_товар а	Код_поставщи ка
1	1	1	2	1	2
1	2	2	1	2	1
2	1	1	1	1	1

Нормальные формы

1НФ - все атрибуты отношения простые;

2НФ - отношение находится в 1НФ и не содержит частичных ФЗ;

3НФ - отношение находится во 2НФ и не содержит *транзитивных* ФЗ от ключа;

НФБК - отношение находится в 3НФ и не содержит ФЗ ключей от неключевых атрибутов;

4НФ, применяется при наличии более чем одной многозначной ФЗ - отношение находится в НФБК или 3НФ и не содержит независимых многозначных ФЗ;

5НФ - отношение находится в 4НФ и не содержит ФЗ по соединению.

Требования к схеме отношений

- исключать избыточное дублирование;
- исключать потенциальную противоречивость данных;
- обладать свойством соединения без потерь;
- обладать свойством сохранения ФЗ.

Методы проектирования логической модели реляционной БД

- Метод декомпозиции отношений
- Метод синтеза отношений

Универсальное отношение

Универсальное отношение – это отношение, которое включает все атрибуты из предметной области базы данных.

Универсальное отношение находится в 1НФ

Пример универсального отношения

Алгоритм метода декомпозиции

- Разработка *универсального отношения* для базы данных.
- Определение всех ФЗ между атрибутами отношения.
- Определение, находится ли отношение в НФБК. Если да, то завершить проектирование; в противном случае отношение должно быть разбито на два других отношения.
- Повторение пунктов 2 и 3 для каждого нового отношения, полученного в результате декомпозиции.

Пример

Минимальное покрытие

Алгоритм метода синтеза

Пример

Правила

Правила преобразования ER-диаграмм в отношения базы данных

Правило 1

Если степень *бинарной связи* 1:1 и класс принадлежности обеих сущностей является обязательным, то требуется построение только одного отношения. При этом *первичным ключом отношения* может быть ключ любой сущности.

Пример

Правило 2

Если степень *бинарной связи* 1:1 и класс принадлежности одной сущности является обязательным, а другой сущности - не обязательным, то требуется построение двух отношений - по одному на каждую сущность. При этом первичным ключом каждого отношения является ключ его сущности, а ключ сущности с необязательным классом принадлежности добавляется в отношение для сущности с обязательным классом принадлежности в качестве атрибута (миграция ключа).

Пример

Правило 3

Если степень *бинарной связи* 1:1 и класс принадлежности обеих сущностей не является обязательным, то требуется построение трех отношений - по одному на каждую объектную сущность и одному для связывающего отношения. При этом ключ каждой сущности является первичным ключом соответствующего отношения и одного отношения для связи, с первичным ключом, составленным из ключей объектных сущностей.

Пример

Правило 4

Если степень *бинарной связи* 1:N, и класс принадлежности n-связной сущности является обязательным, то достаточно построить два отношения - по одному на каждую сущность. При этом ключ каждой сущности является первичным ключом соответствующего отношения, а ключ 1-связной сущности добавляется в отношение для n-связной сущности в качестве атрибута.

Пример

Правило 5

Если степень *бинарной связи* 1:N и класс принадлежности n-связной сущности не является обязательным, то необходимо построить три отношения - по одному на каждую сущность. При этом ключ каждой сущности является первичным ключом соответствующего отношения и одного отношения для связи. Ключи сущностей должны быть атрибутами последнего отношения.

Пример

Правило 6

Если степень *бинарной связи* M:N, то необходимо построить три отношения - по одному для каждой сущности и одно отношение для связи. При этом ключ каждой сущности является *первичным ключом* соответствующего отношения, и входит в составной *первичный ключ отношения* для связи.

Пример

Правило 7

Если связь является трехсторонней, необходимо построить четыре отношения - по одному на каждую сущность. При этом ключ каждой сущности является первичным ключом соответствующего отношения и одного отношения для связи. Ключи сущностей должны быть атрибутами последнего отношения.

Пример

Правило 8

Исходная сущность служит источником генерации одного отношения. Ключ сущности есть ключ *отношения*.

Подчиненные сущности (ролевые элементы) и связи, соединяющие их, порождают такое количество отношений, которое определяется набором правил 1-7, причем каждый ролевой элемент трактуется как обычная сущность.