

# DWH

Лекция №3  
Проектирование схемы БД по схеме Data Vault,  
Anchor modeling.



В прошлой  
лекции



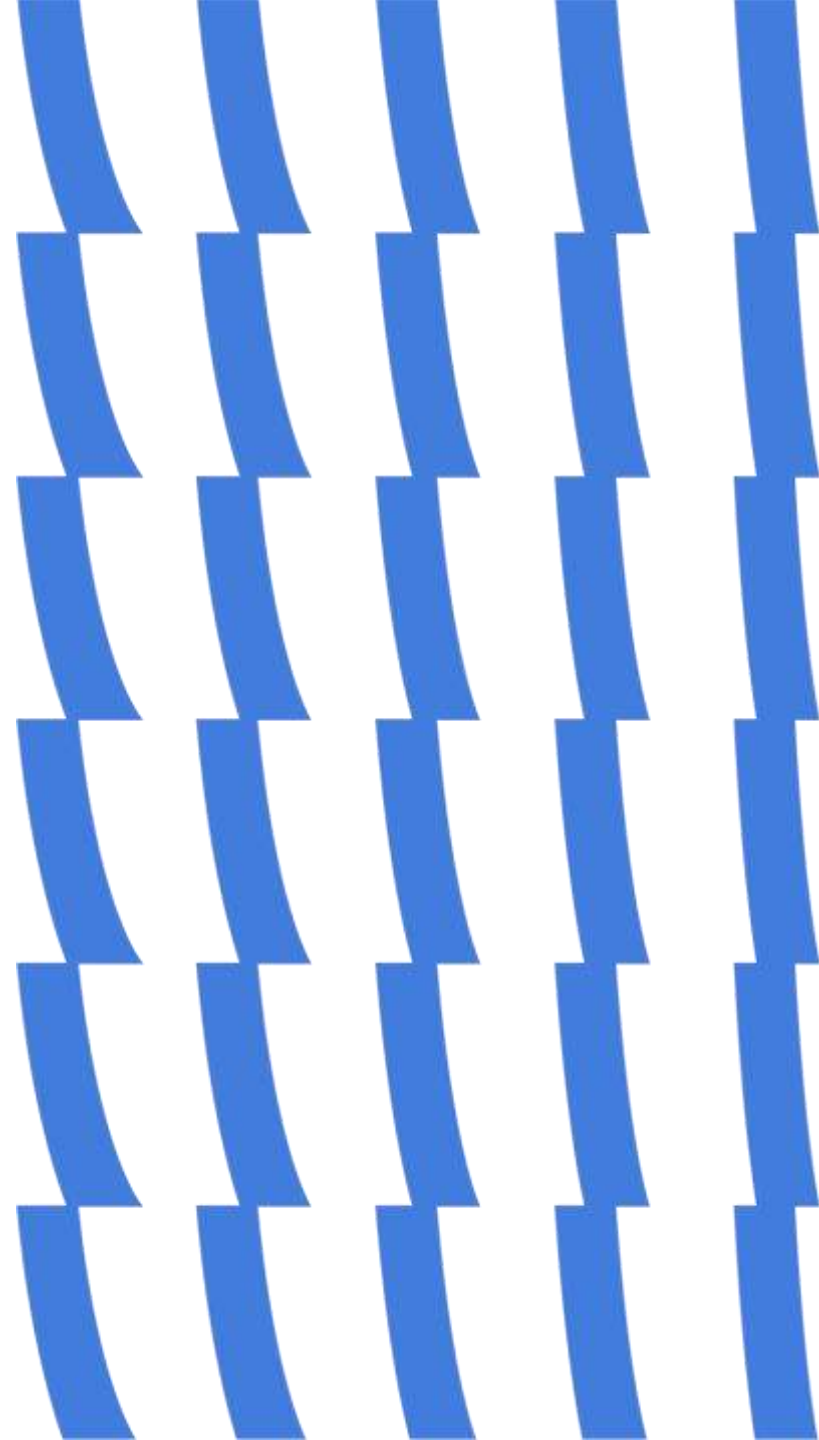
# Реляционная модель данных

Каждая БД и СУБД строится на основе некоторой явной или неявной модели данных. Все СУБД, построенные на одной и той же модели данных, относят к одному типу.

Основой реляционных СУБД является реляционная модель данных (РМД).

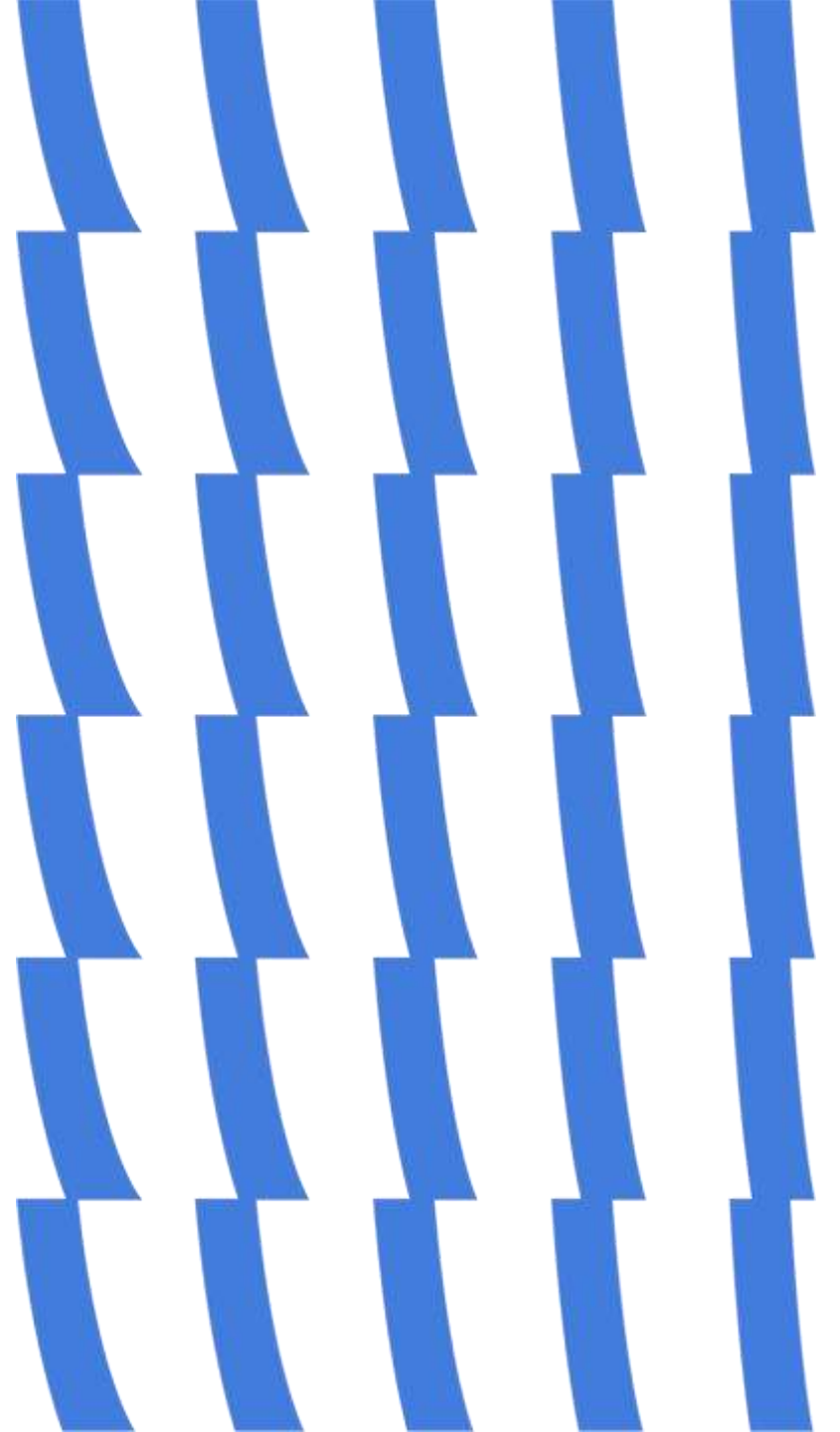
Три аспекта этой модели:

- Структура — Что за объекты? Данные в базе данных представляют собой набор отношений.
- Целостность — Как определяются корректные состояния базы? Ограничения целостности в РМД задаются на уровне домена, отношения или всей БД.
- Манипулирование — Способы модификации БД. Операторы модификации описываются реляционной алгеброй.



# Нормализация

- Нормализация - процесс декомпозиции отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы
- 1НФ, 2НФ — денормализованные отношения, используются в хранилищах данных для ускорения доступа
- 3НФ, НФБК — классические модели по схеме снежинка\звезда  
Отношение находится в третьей нормальной форме (3НФ) тогда и только тогда, когда оно находится во второй нормальной форме и каждый неключевой атрибут полностью функционально зависит только от ключей.
- НФБК, 4НФ — подход Data Vault
- 5НФ, 6НФ — подход Anchor modelling



# «СУЩНОСТЬ-СВЯЗЬ»

Модель данных или ER-модель (от Entity relationship model — «сущность-связь»): набор взаимосвязанных сущностей, которые сгруппированы по функциональным областям. Разрабатывается последовательно и состоит из:

1. Концептуальной модели
2. Логической модели
3. Физической модели



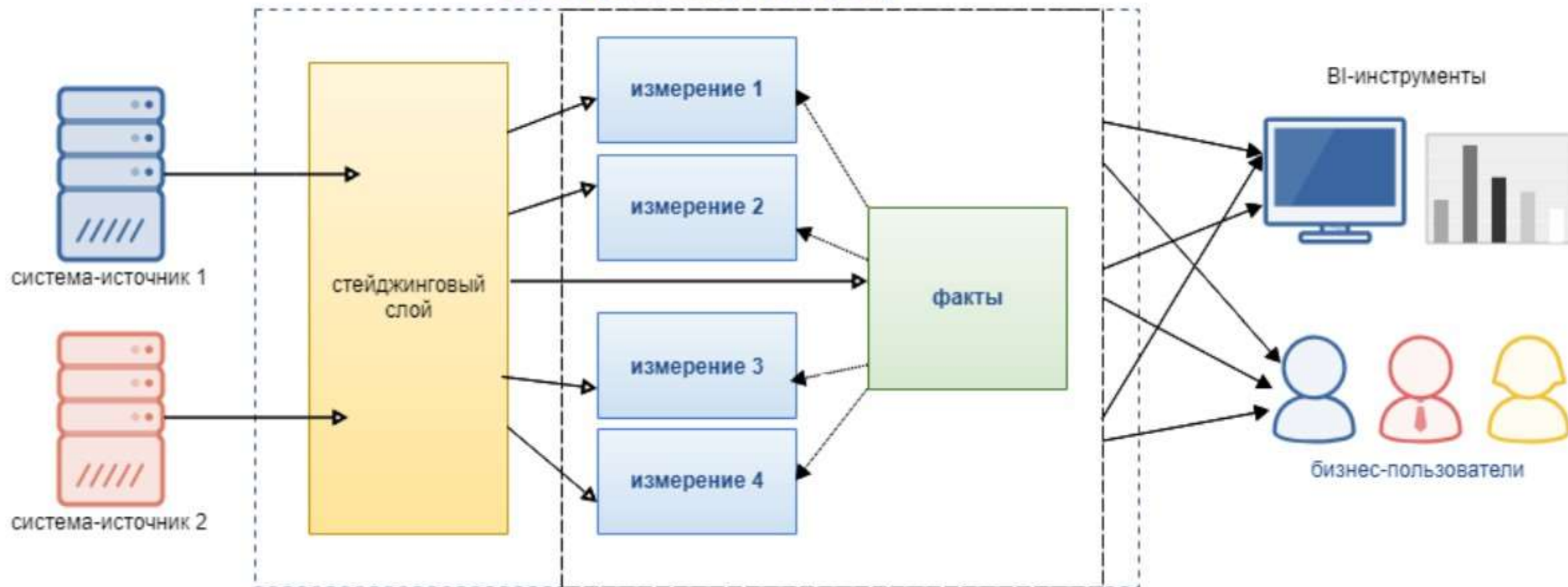
# Подход Инмона



STG -> DDS -> EMART

# Подход Кимбалла

Схема хранилища по Кимбаллу



DWH по Кимбаллу - набор витрин данных.

Детального слоя в понимании Инмона в этой модели нет, поэтому первые результаты можно получить достаточно быстро.



# Многомерное моделирование





# Многомерное моделирование

Многомерное моделирование (Dimensional modeling) является методом моделирования и визуализации данных как множества числовых или лингвистических показателей или параметров (measures), которые описывают общие аспекты деятельности организации. Как правило, при многомерном моделировании основное внимание фокусируется на числовых данных, таких как число продаж, баланс, прибыль, вес, или на объектах, которые можно пересчитать, таких как статьи, патенты, книги.

В многомерных моделях данные структурированы таким образом, чтобы оптимизировать обработку запросов к базе данных и анализ при работе с большими объемами данных.

Многомерное моделирование имеет много общего с моделированием методом "сущность-связь" для реляционной модели, но отличается целями. Реляционная модель акцентируется на целостности и эффективности ввода данных. Многомерная модель (Dimensional model) ориентирована в первую очередь на выполнение сложных запросов к БД.



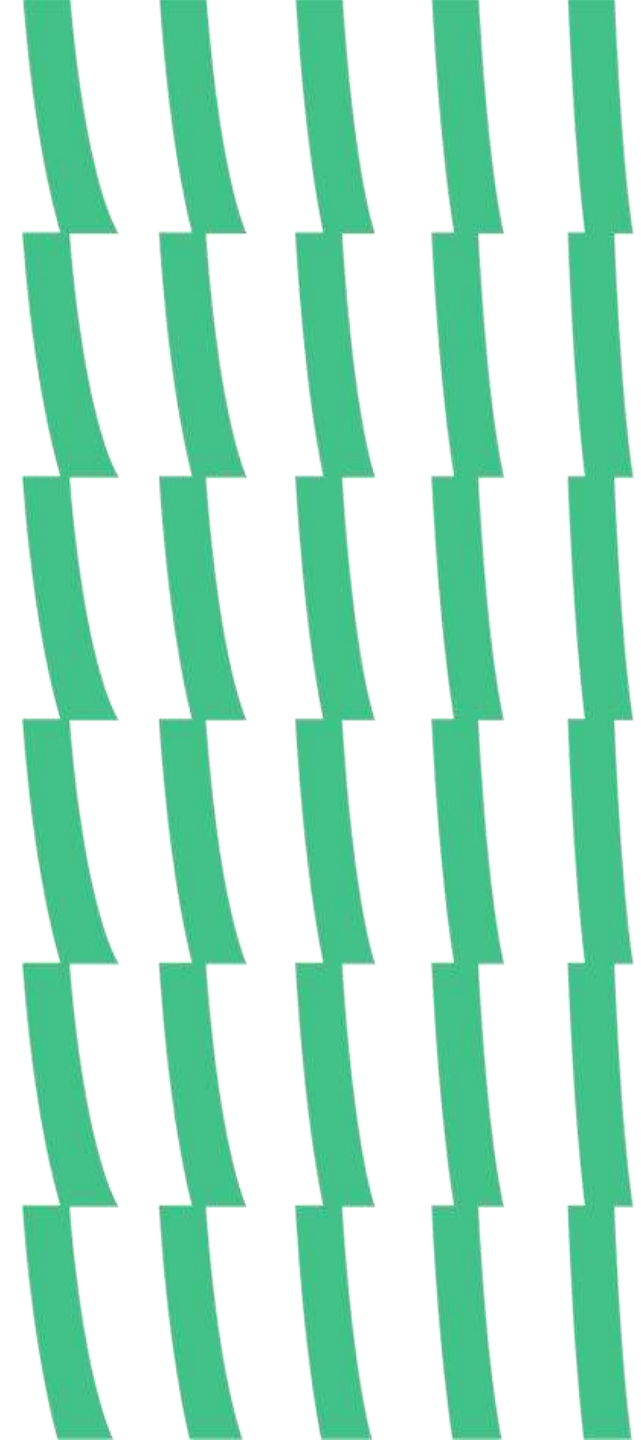
# Многомерное моделирование

Метод многомерного моделирования базируется на следующих основных понятиях: факты, атрибуты, измерения, параметры (метрики), иерархия, гранулированность.

**Факт (fact)** — это набор связанных элементов данных, содержащих метрики и описательные данные. Каждый факт обычно представляет элемент данных, численно описывающий деятельность организации, бизнес-операцию или событие, которое может быть использовано для анализа деятельности организации или бизнес-процессов. В ХД факты сохраняются в базовых таблицах реляционной БД. Например, стоимость товара, количество единиц товара и т.д.

**Параметр, метрика или показатель (measure)** — это числовая характеристика факта, который определяет эффективность деятельности или бизнес-действия организации с точки зрения измерения. Как правило, метрика содержит заранее не известное значение характеристики факта. Конкретные значения метрики описываются с помощью переменных. Например, пусть метрикой является численное выражение продаж товара в деньгах, количество проданных единиц товара и т.д. Метрика определяется с помощью комбинации элементов измерения и, таким образом, представляет факт.

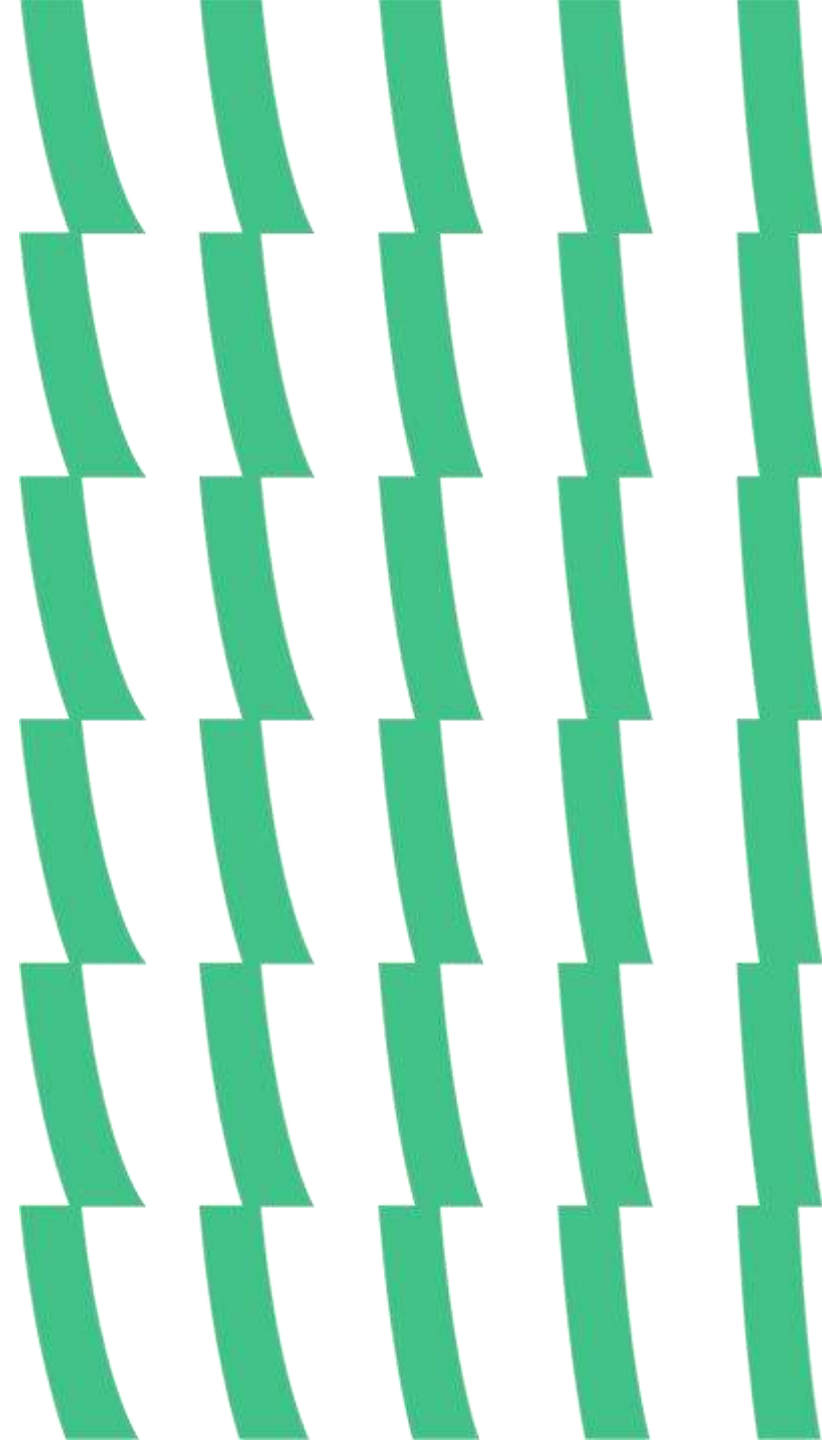
**Гранулированность (Granularity)** - это уровень детализации данных, сохраняемых в ХД, степень подробности (детализации) описания факта в строке таблицы фактов. Например, в многомерной модели, описывающей процесс записи студентов на обучающие курсы, уровень гранулированности можно определить так: студент, дата, курс



# Многомерное моделирование

**Измерение (dimension)** — это интерпретация факта с некоторой точки зрения в реальном мире. Измерения, подобно атрибутам, содержат текстовые значения, которые сильно связаны по смыслу между собой. Обычно измерения представляются как оси многомерного пространства, точками которого являются связанные с ними факты. В многомерной модели каждый факт связан с одной или несколькими осями. Измерения обычно представляют нечисловые, лингвистические переменные, такие как филиалы организации, сотрудники организации, покупатели и т.д.

**Атрибут (Attribute)** - это описание характеристики реального объекта предметной области. Как правило, атрибут содержит заранее известное значение, характеризующее факт. Обычно атрибуты представляются текстовыми полями с дискретными значениями. Например, габариты упаковки товара, запах товара.

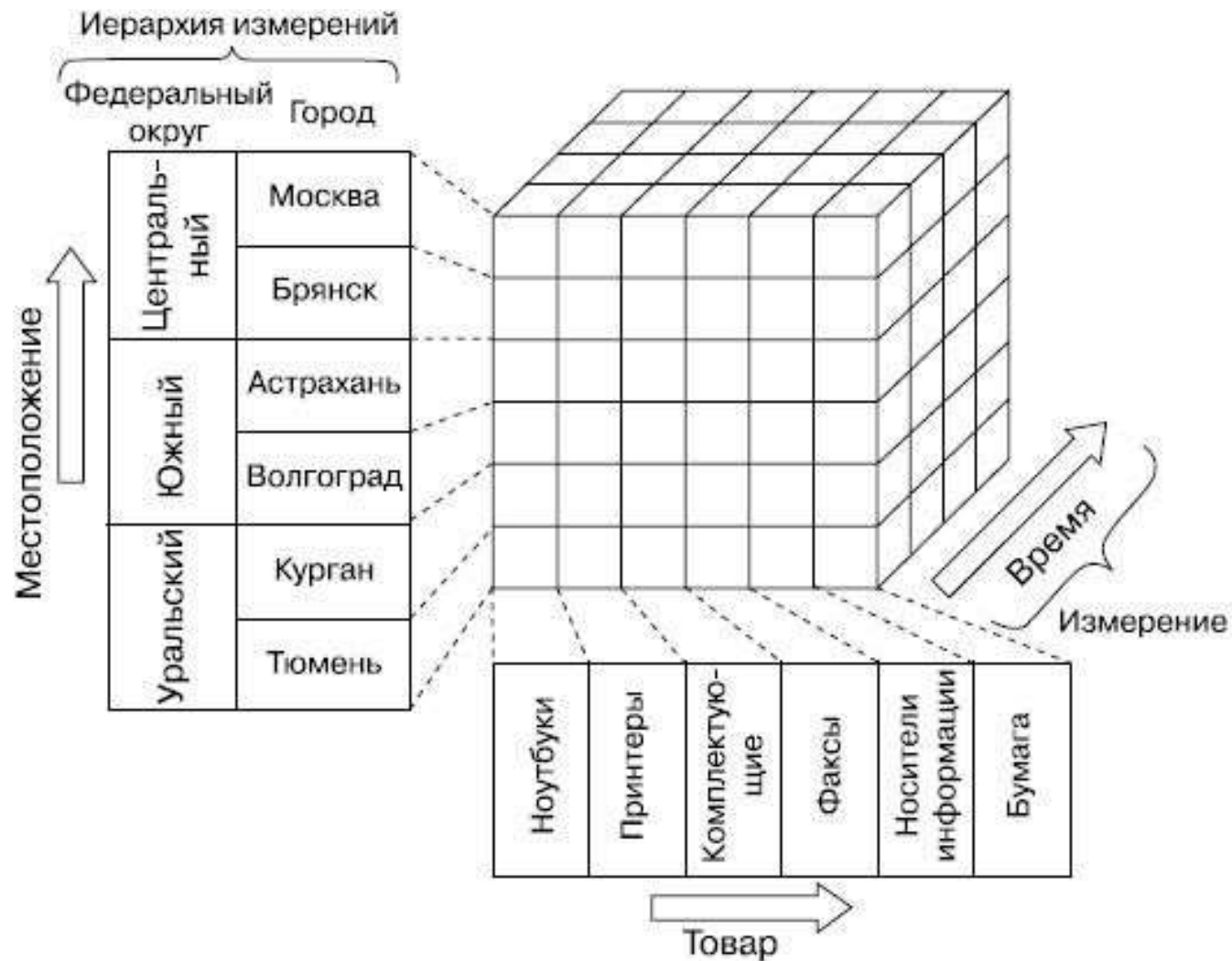


# Многомерное модель

Многомерная модель визуально представляется с помощью куба (или в случае более трех измерений – гиперкуба). Рассмотрим пример. Пусть объем продаж торговой организации есть функция от переменных "Товары", "Месяц" и "Регион продаж". Тогда в качестве измерений будут выступать "Товары", "Время" и "Месторасположение".

На этих измерениях могут быть заданы следующие иерархии:

- измерение "Товары" – "Производитель-Категория-Товар";
- измерение "Месторасположение" – "Федеральный округ-Город-Магазин";
- измерение "Время" – "Год-Квартал-Месяц" или "Неделя-День".

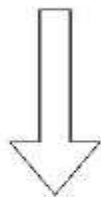


# Многомерное моделирование

Многомерное моделирование является основным методом логического проектирования ХД для OLAP-приложений. Для таких приложений типично выполнение операций свертывания и развертывания данных.

Развертка (drill down) и свертка (drill up) являются операциями перемещения вниз и вверх по уровням иерархии измерения. При выполнении развертки пользователь перемещается от верхних уровней к нижним уровням, которые содержат обычно более подробные данные. При выполнении свертки пользователь перемещается от нижних уровней иерархии к верхним, тем самым обобщая информацию на каждом уровне. При выполнении этих операций путь навигации определяется иерархиями измерений.

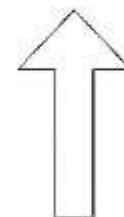
	20.03	20.04	20.05
Центральный	142500	241400	212600
Южный	50200	52600	51400



Развертка (drill down)  
по измерению  
«Регион»

	20.03	20.04	20.05
Москва	131000	220000	200000
Тверь	21500	21400	12600
Ростов	20100	21600	21400
Волгоград	30100	32000	30000

Центральный	212600
Южный	51400



Свертка (drill up)  
по измерению  
«Время»

	1 кв.	2 кв.	3 кв.	4 кв.
Центральный	53600	53150	53000	53150
Южный	12850	12700	2300	12850



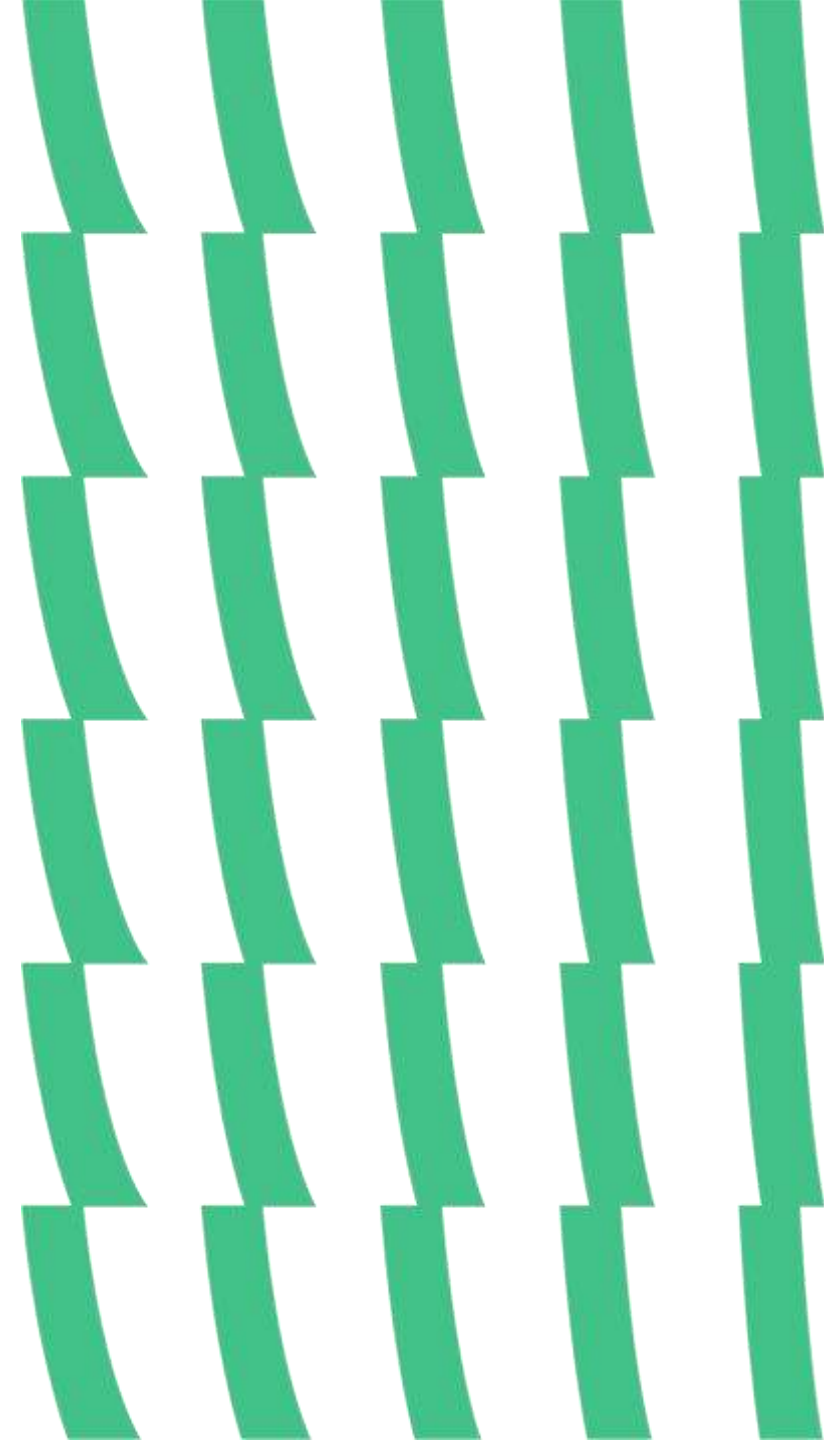
# Факты и измерения

**факты** - численные параметры, данные, которые количественно описывают процесс, они непрерывны, т. е. могут принимать бесконечное количество значений.

**измерения** - текстовые параметры для предоставления максимально возможного контекста для фактов(могут быть числовыми, но в любом случае это дискретные данные).

Основной таблицей хранилища данных является **таблица фактов**. Обычно такие таблицы содержат сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться.

**Таблицы измерений** содержат практически неизменяемые данные.



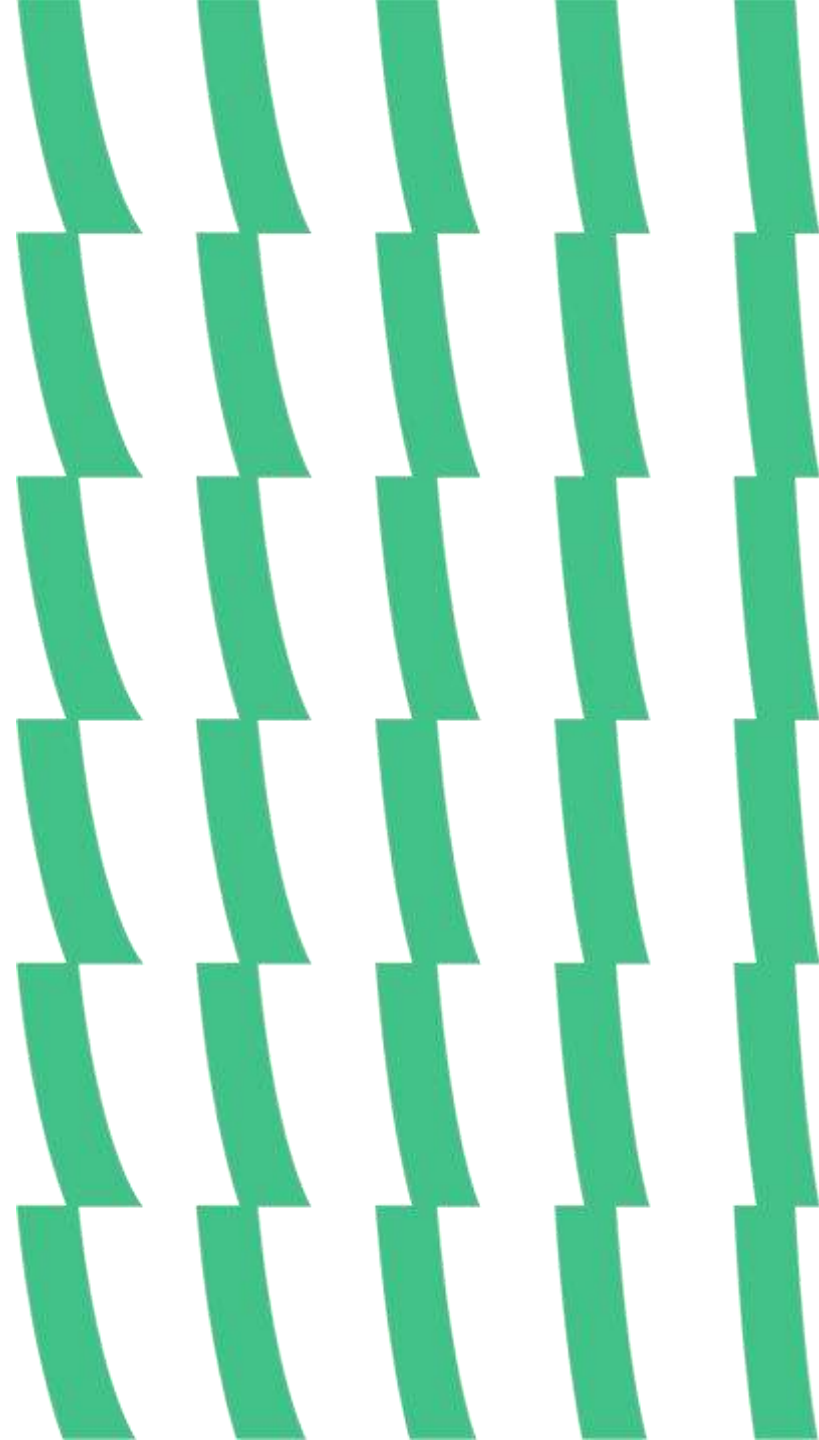


# Таблицы фактов

Таблица **фактов** — является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах, событиях или процессах, совокупность которых будет в дальнейшем анализироваться.

Характеристики таблиц фактов:

- Таблица фактов содержит числовые параметры (метрики);
- Каждая таблица фактов имеет составной ключ, состоящий из первичных ключей таблиц измерений. Первичный ключ таблицы измерений является внешним ключом в таблице фактов.



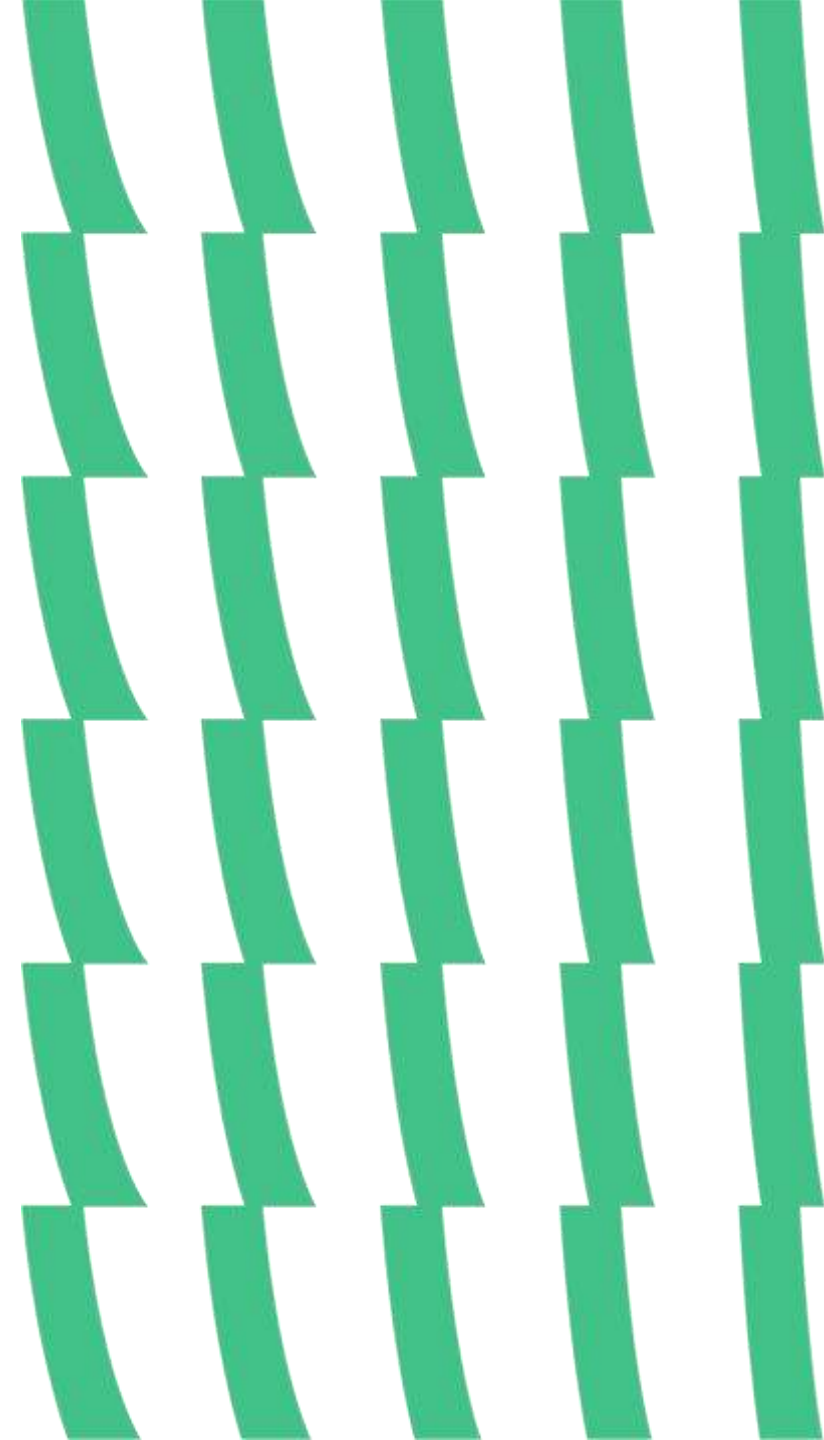
## Пример таблицы фактов

The screenshot shows the SQL Server Enterprise Manager interface. The title bar reads "SQL Server Enterprise Manager - [3:Data in Table 'Sales\_Fact' in 'Northwind\_Mart' on 'MAINDESK']". The menu bar includes "Console", "Window", and "Help". The toolbar contains various icons for database operations. The main area displays a table with the following columns: TimeKey, CustomerKey, ShipperKey, ProductKey, EmployeeKey, RequiredDate, LineItemFreight, LineItemTotal, LineItemQuantity, and LineItemDiscount. The table contains 15 rows of data, with the first row highlighted. The data is as follows:

TimeKey	CustomerKey	ShipperKey	ProductKey	EmployeeKey	RequiredDate	LineItemFreight	LineItemTotal	LineItemQuantity	LineItemDiscount
3	85	4	11	5	01.08.1996	14.3904	168	12	0
5	85	4	42	5	01.08.1996	11.992	98	10	0
5	85	4	72	5	01.08.1996	5.996	174	5	0
1	79	1	14	6	16.08.1996	2.1321	167.4	9	0
1	79	1	51	6	16.08.1996	9.476	1696	40	0
3	34	2	41	4	05.08.1996	10.971	77	10	0
3	34	2	51	4	05.08.1996	38.3985	1484	35	222.6
3	34	2	65	4	05.08.1996	16.4565	252	15	37.8
4	84	1	22	3	05.08.1996	6.0492	100.8	6	5.04
4	84	1	57	3	05.08.1996	15.123	234	15	11.7
4	84	1	65	3	05.08.1996	20.164	336	20	0
2	76	2	20	4	06.08.1996	19.54	2592	40	129.6
2	76	2	33	4	06.08.1996	12.2125	50	25	2.5
2	76	2	60	4	06.08.1996	19.54	1088	40	0
5	34	2	31	3	24.07.1996	11.404	200	20	0

# Категории фактов

1. события (event), по крайней мере, на уровне самой большой гранулярности, как правило, моделируют события реального мира, при этом каждый факт представляет определенный экземпляр изучаемого явления. Примерами могут служить продажи, щелчки мышью на веб-странице или движение товаров на складе;
2. мгновенные снимки (snapshot) моделируют состояние объекта в данный момент времени, такие как уровни наличия товаров в магазине или на складе и число пользователей веб-сайта. Один и тот же экземпляр явления реального мира, например, конкретная банка бобов, может возникать в нескольких фактах;
3. совокупные мгновенные снимки (cumulative snapshot) содержат информацию о деятельности организации за определенный отрезок времени. Например, совокупный объем продаж за предыдущий период, включая текущий месяц, можно легко сравнить с показателями за соответствующие месяцы прошлого года.

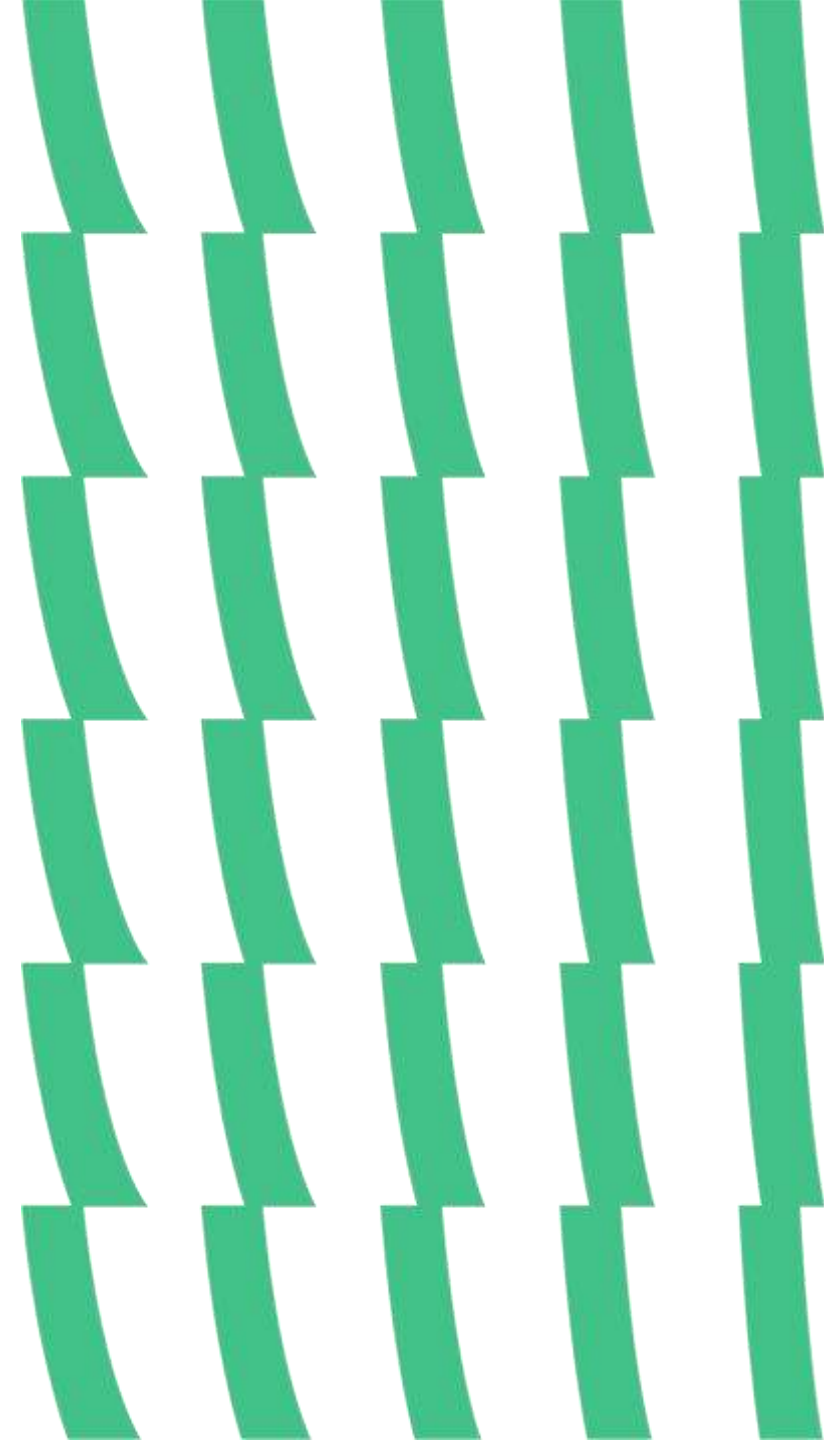


# Сравнение видов таблиц фактов

	Транзакционная таблица фактов	Таблица фактов периодических моментальных снимков	Таблица фактов кумулятивных моментальных снимков
Определение гранулярности таблицы фактов	Одна строка на бизнес-операцию	Одна строка на период	Одна строка для периода завершенного события
Измерения	Используются факты на самом низком уровне детализации «дата/время»	Используются факты на конец периода измерения «дата/время»	Используются факты по нескольким измерениям «дата/время» для фиксации результатов в различных контрольных точках
Факты	Факты связаны с операционной деятельностью	Факты связаны с периодической деятельностью	Факты связаны с деятельностью, имеющей определенное время существования
Количество записей таблицы фактов	Растет быстро	Растет медленнее, чем в транзакционных таблицах	Растет медленнее всего
Обновления	Не допускаются	Не допускаются	Допускаются

# Виды таблиц фактов

- **Аддитивные факты** (Additive facts). Факт называется аддитивным, если его имеет смысл использовать с любыми измерениями для выполнения операций суммирования с целью получения какого-либо значимого результата. Например, количество продаж, объем продаж и т.д.;
- **Полуаддитивные факты** (Semiadditive facts). Факт называется полуаддитивным, если его имеет смысл использовать совместно с некоторыми измерениями для выполнения операций суммирования с целью получения какого-либо значимого результата. Например, числовые показатели интенсивности, такие как остаток на счете, уровень запасов на складе и т.д.;
- **Неаддитивные факты** (Non-additive facts). Факт называется неаддитивным, если его не имеет смысла использовать совместно с каким-либо измерением для выполнения операций суммирования с целью получения какого-либо значимого результата. Например, измерение комнатной температуры.





# Аддитивные факты

Аддитивные факты можно суммировать по всем измерениям. Количество продаж и суммарная прибыль - аддитивные факты.

Например, имеет смысл суммировать общий объем продаж для продукта, местоположения и времени, поскольку это не вызывает наложения среди явлений реального мира, которые генерируют каждое из этих значений.

По измерению "Время":

Дата	Товар	Магазин	Количество продаж	Количество покупателей	Суммарная прибыль
23.01.2009	CD диск	Компьютер	10	10	1500
24.01.2009	CD диск	Компьютер	35	30	5250
25.01.2009	CD диск	Компьютер	20	15	3000
			<b>65</b>	<b>55</b>	<b>9750</b>

По измерению "Товар":

Дата	Товар	Магазин	Количество продаж	Количество покупателей	Суммарная прибыль
23.01.2009	CD диск	Компьютер	10	6	1500
23.01.2009	Принтер	Компьютер	1	1	5000
23.01.2009	Сканер	Компьютер	2	2	3000
			<b>13</b>		<b>9500</b>

По измерению "Магазин":

Дата	Товар	Магазин	Количество продаж	Количество покупателей	Суммарная прибыль
23.01.2009	CD диск	Компьютер	10	10	1500
23.01.2009	CD диск	Принтеры	10	5	1500
23.01.2009	CD диск	Оргтехника	20	7	3000
			<b>40</b>	<b>22</b>	<b>6000</b>



# Полуаддитивные факты

Полуаддитивные факты не имеет смысла суммировать по некоторым измерениям.

Количество покупателей - полуаддитивный факт

Зададим следующий вопрос: "Сколько покупателей купили либо бумагу для принтера, либо бумагу для факса?". Ответ: где-то между 7-ю и 13-ю. Невозможность разделить количество покупателей между товарами для приведенной сущности делает этот факт *полуаддитивным* для измерения "Товары".

Суммирование метрики "Количество покупателей" по измерению "Товары":

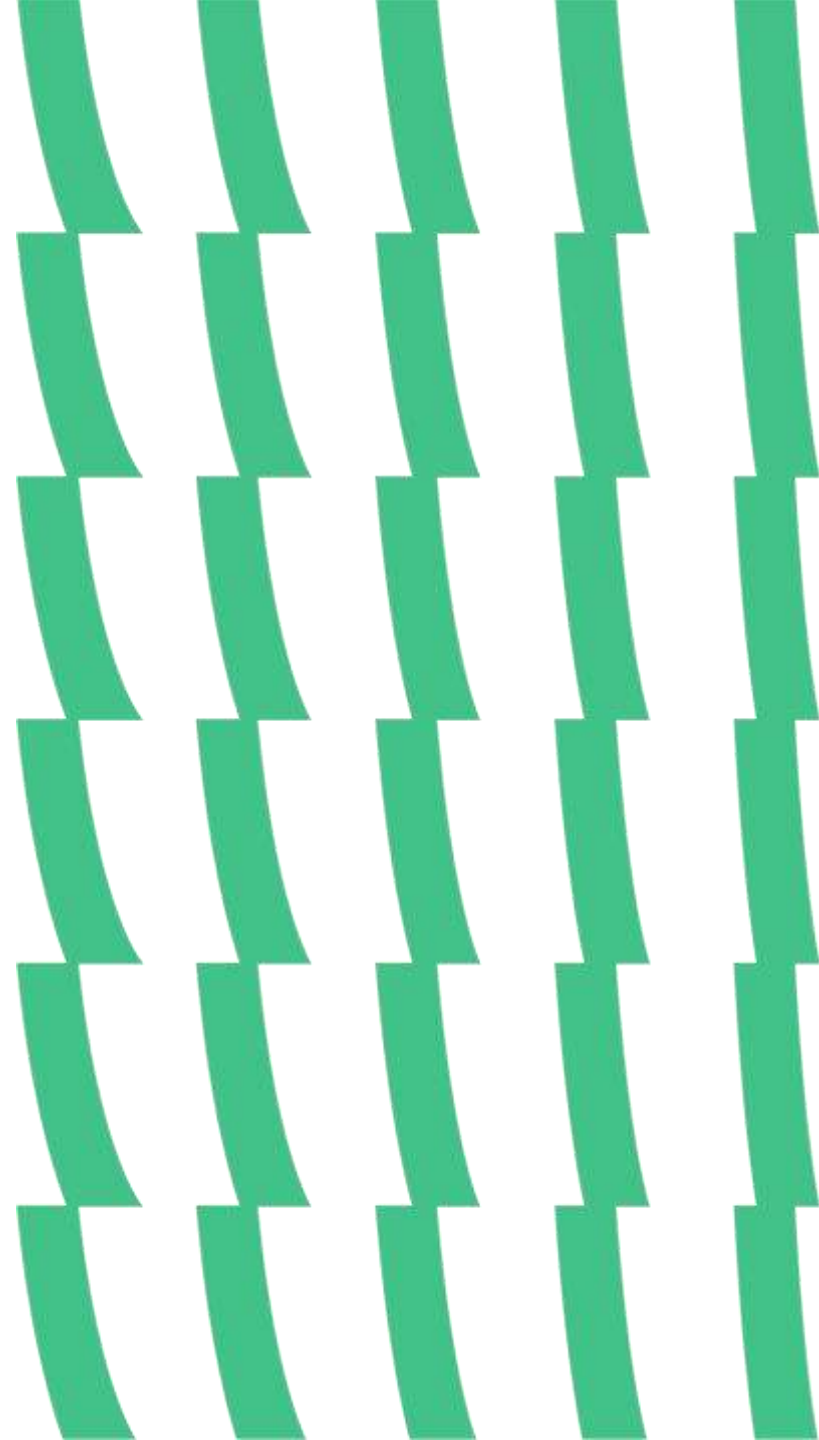
Дата	Товар	Магазин	Количество продаж	Количество покупателей	Суммарная прибыль
23.01.2009	Бумага для факсов	Компьютер	10	6	1000
23.01.2009	Бумага для принтера	Компьютер	12	7	1320
				13	

# Неаддитивные факты

Неаддитивные факты не имеет смысла суммировать ни по каким измерениям.

Проценты и отношения величин (конверсия посетителей в покупателей) являются неаддитивными фактами. Можно хранить как отдельные параметры числитель и знаменатель отношения, когда их раздельное суммирование имеет смысл. И это будут уже аддитивные факты. К неаддитивным фактам относятся также статистические средние суммы, средняя температура за день. Сумма средних дневных температур за неделю не имеет никакого смысла.

Неаддитивные параметры не комбинируются в любом измерении обычно потому, что выбранная формула не позволяет объединить средние значения низкого уровня в среднем значении более высокого уровня. Аддитивные и неаддитивные параметры могут описывать факты любого рода, в то время как полуаддитивные параметры, как правило, используются с мгновенными снимками или совокупными мгновенными снимками.



# Свойства таблиц фактов

Свойства данных в таблицах фактов:

- числовые параметры используются для агрегации и суммирования;
- значения данных должны обладать свойствами аддитивности или полуаддитивности по отношению к измерениям, для того чтобы их можно было суммировать;
- все данные таблицы фактов должны быть однозначно идентифицированы через ключи таблиц измерений, чтобы обеспечить доступ к ним через таблицы измерений;

Таким образом, таблицу фактов можно разделить на две части. Первая часть состоит из первичных ключей измерений, а вторая — из числовых параметров функционально зависящих от ключей таблиц измерений.



# Измерения

Таблица измерений (англ. dimension table) — таблица в структуре многомерной базы данных, которая содержит атрибуты событий, сохраненных в таблице фактов. Атрибуты представляют собой текстовые или иные описания, логически объединенные в одно целое.

Таблица измерения имеет первичный ключ и атрибуты, описывающие факты с точки зрения некоторого направления деятельности организации.

Характеристики измерений:

- Таблицы измерений содержат данные о детализации фактов;
- Таблицы измерений содержат описательную информацию о числовых значениях в таблице фактов, т.е. они содержат атрибуты фактов;
- Как правило, таблицы измерений содержат большое количество полей;
- Таблицы измерений содержат обычно значительно меньше строк, чем таблицы фактов;
- Атрибуты таблиц измерений обычно используются при визуализации данных в отчетах и запросах;



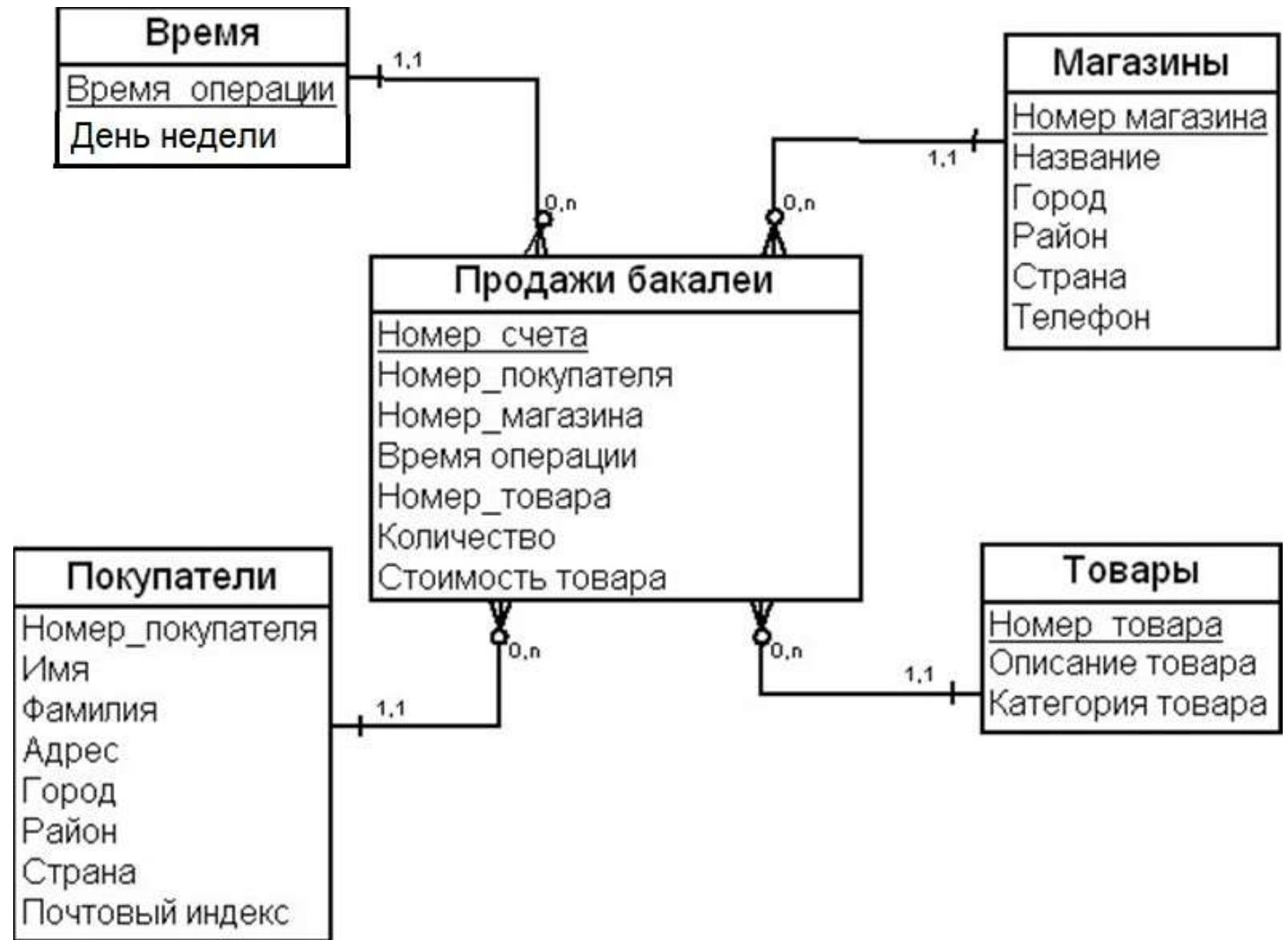
# Модель «Звезда»

Существуют несколько схем для многомерного моделирования данных. Две из них считаются основными: схема "звезда" (star schema) и схема "снежинка" (snowflake schema).

Модель Звезда - другое название: пространственная модель, модель измерений и фактов, модель “сущность-связь”, dimensional model, star schema

Придумал: Ральф Кимбалл

Модель представляется двумя видами таблиц - таблицами **фактов** и таблицами **измерений**, которые описывают факты.

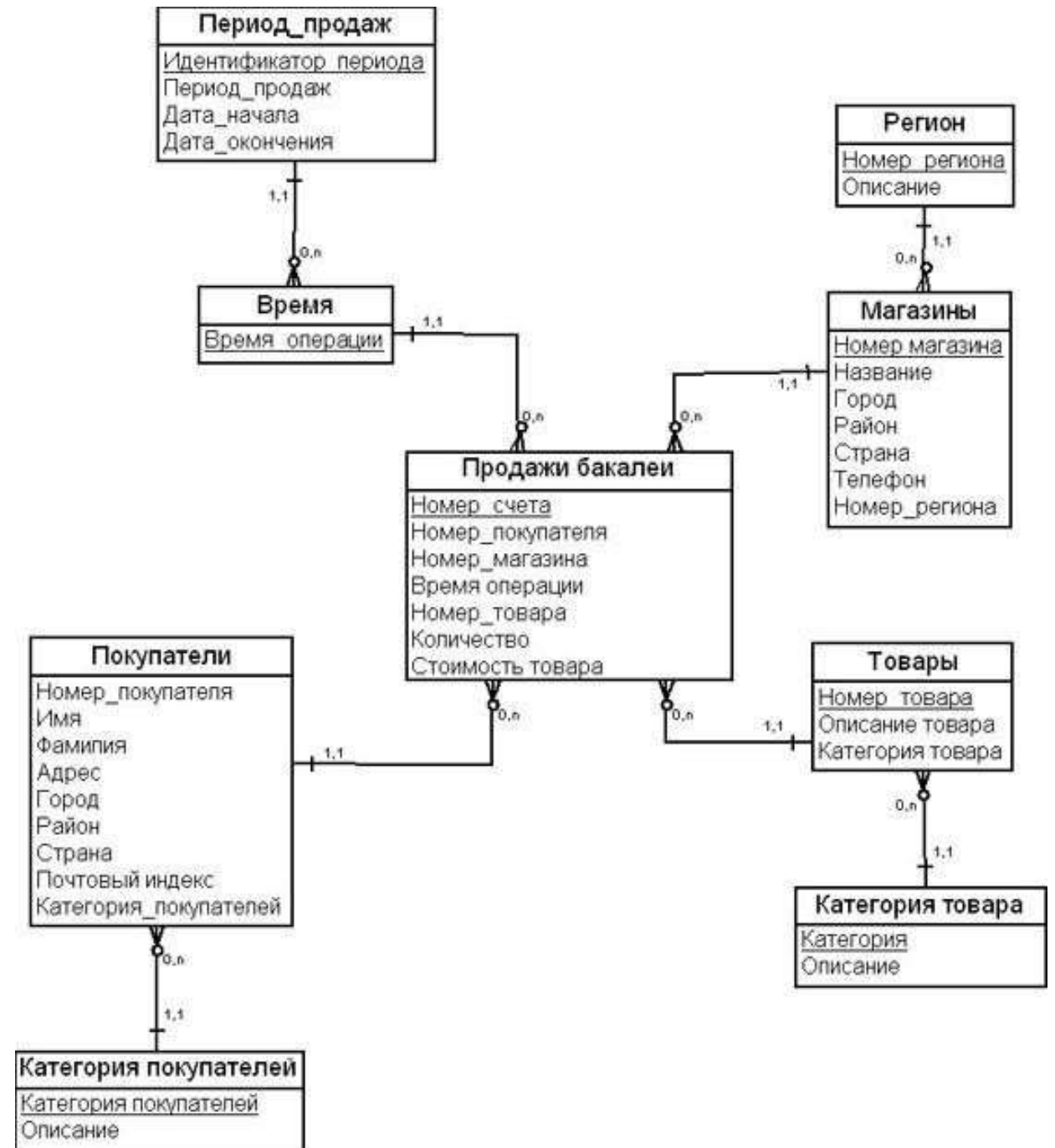




# Модель «Снежинка»

Схема "снежинка" добавляет **иерархию** в таблицы измерений.

Например, измерение "Регион" группирует магазины по географическим регионам, измерение "Категория товара" группирует товары по категориям, измерение "Категория покупателей" группирует покупателей по категориям, а измерение "Период продаж" группирует продажи по периодам времени.



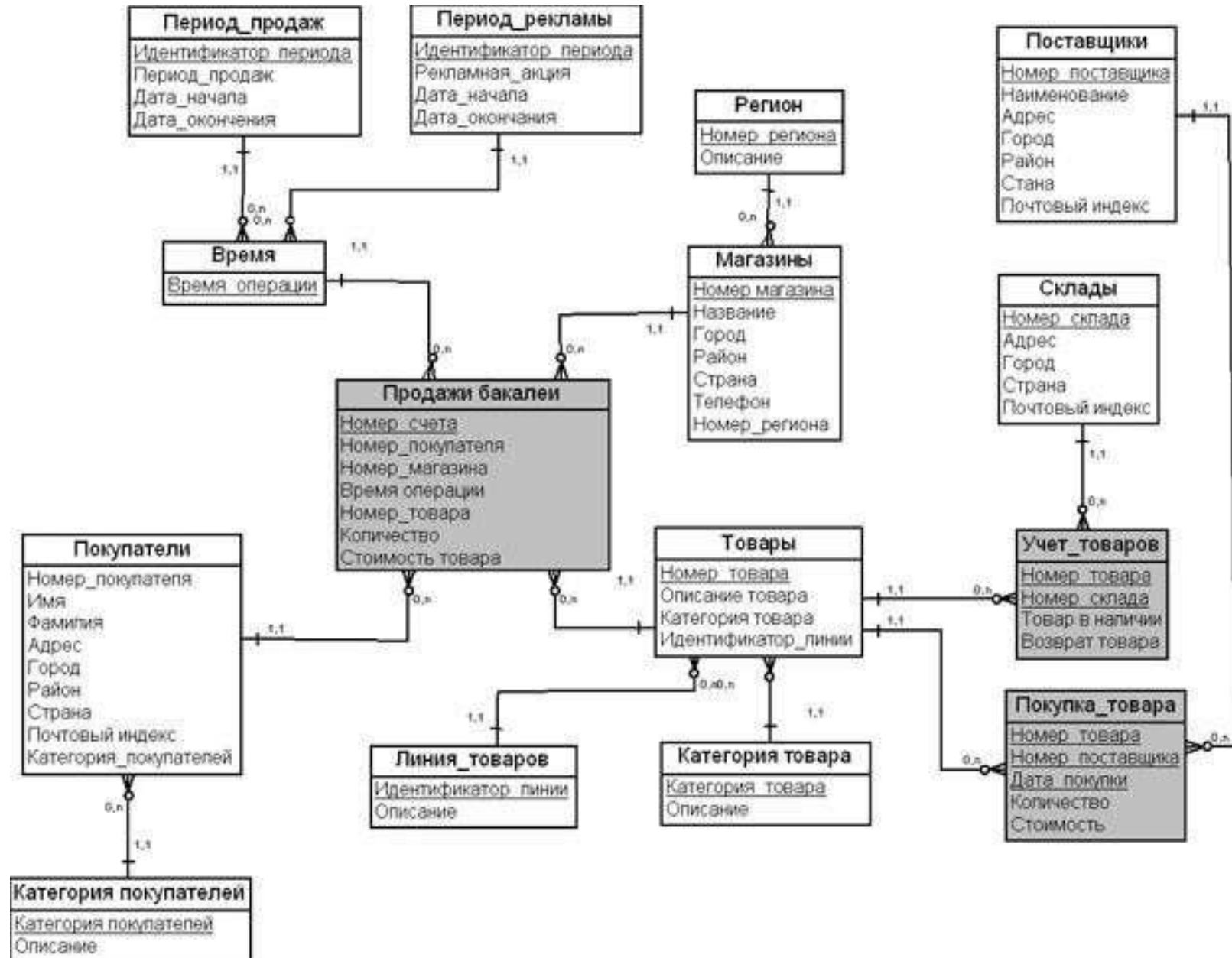


# Схема «звезда против снежинки»: основные отличия

Схема звезды	Схема снежинки
Иерархии для измерений хранятся в таблице измерений.	Иерархии разделены на отдельные таблицы.
Содержит таблицу фактов, окруженную таблицами измерений.	Одна таблица фактов, окруженная таблицей измерений, которая в свою очередь окружена таблицей измерений
В схеме типа «звезда» только одно соединение создает связь между таблицей фактов и любыми таблицами измерений.	Схема снежинки требует много соединений для извлечения данных.
Простой дизайн БД.	Сложный дизайн БД.
Денормализованная структура данных и запрос также выполняются быстрее.	Нормализованная структура данных.
Высокий уровень избыточности данных	Ниже уровень избыточности данных

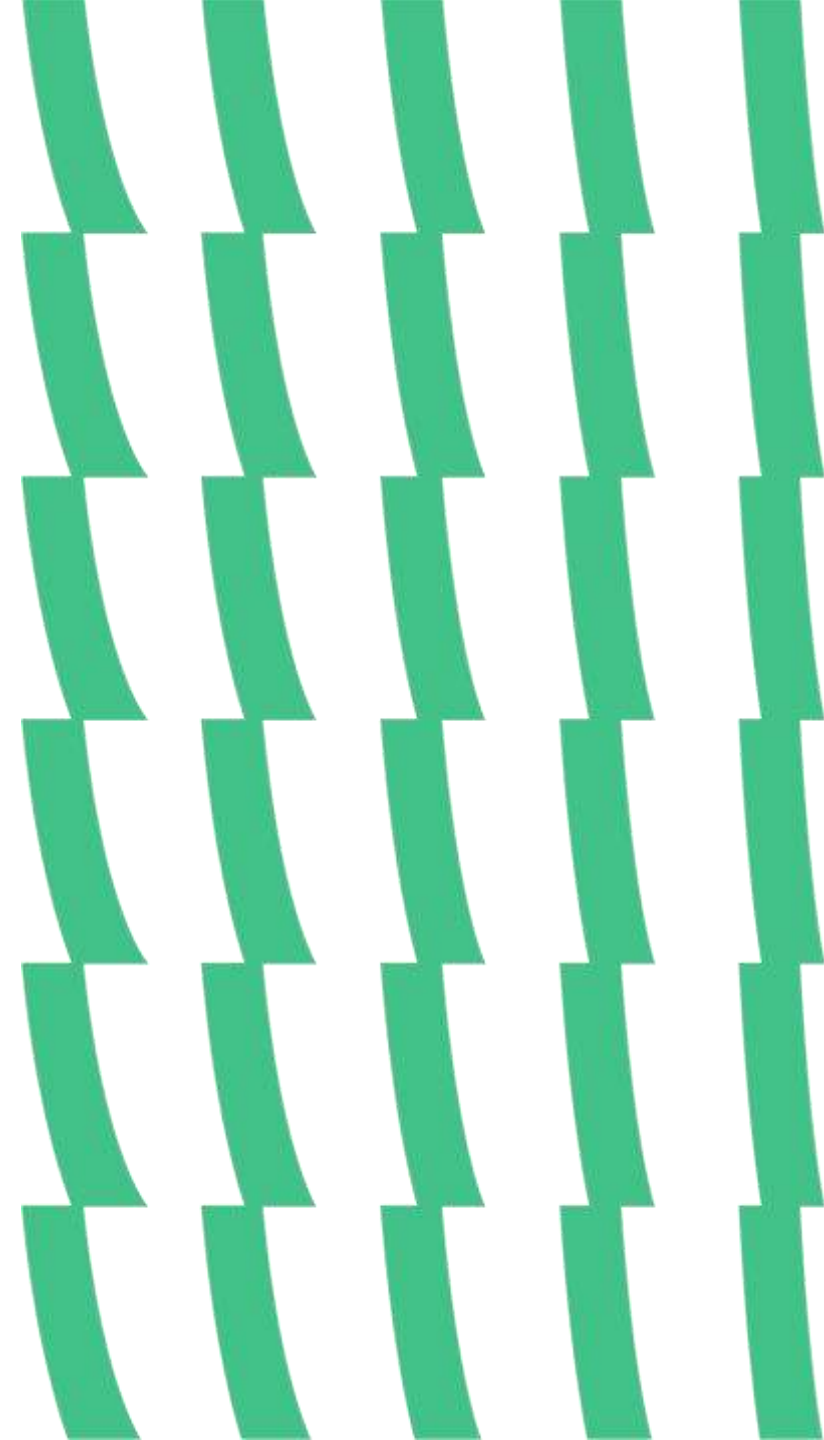
# Модель «Созвездие»

Модель «Созвездие»  
получается из нескольких  
таблиц фактов, которые  
соединяются между собой  
по той или иной логике.



# Виды измерений

- **Медленно меняющимися измерениями** (Slowly Changing Dimensions) называются таблицы измерений, в которых некоторые атрибуты могут изменить свои значения по истечении некоторого периода времени, причем частота таких изменений является небольшой. Всего существует 6 основных типов SCD, которые определяют, как история изменений может быть отражена в модели.
- **Быстро меняющимися измерениями** (Rapidly Changing Dimensions) называются таблицы измерений, в которых некоторые атрибуты могут часто менять свои значения в короткие периоды времени.



# Медленно меняющиеся измерения

## Типы 0-1

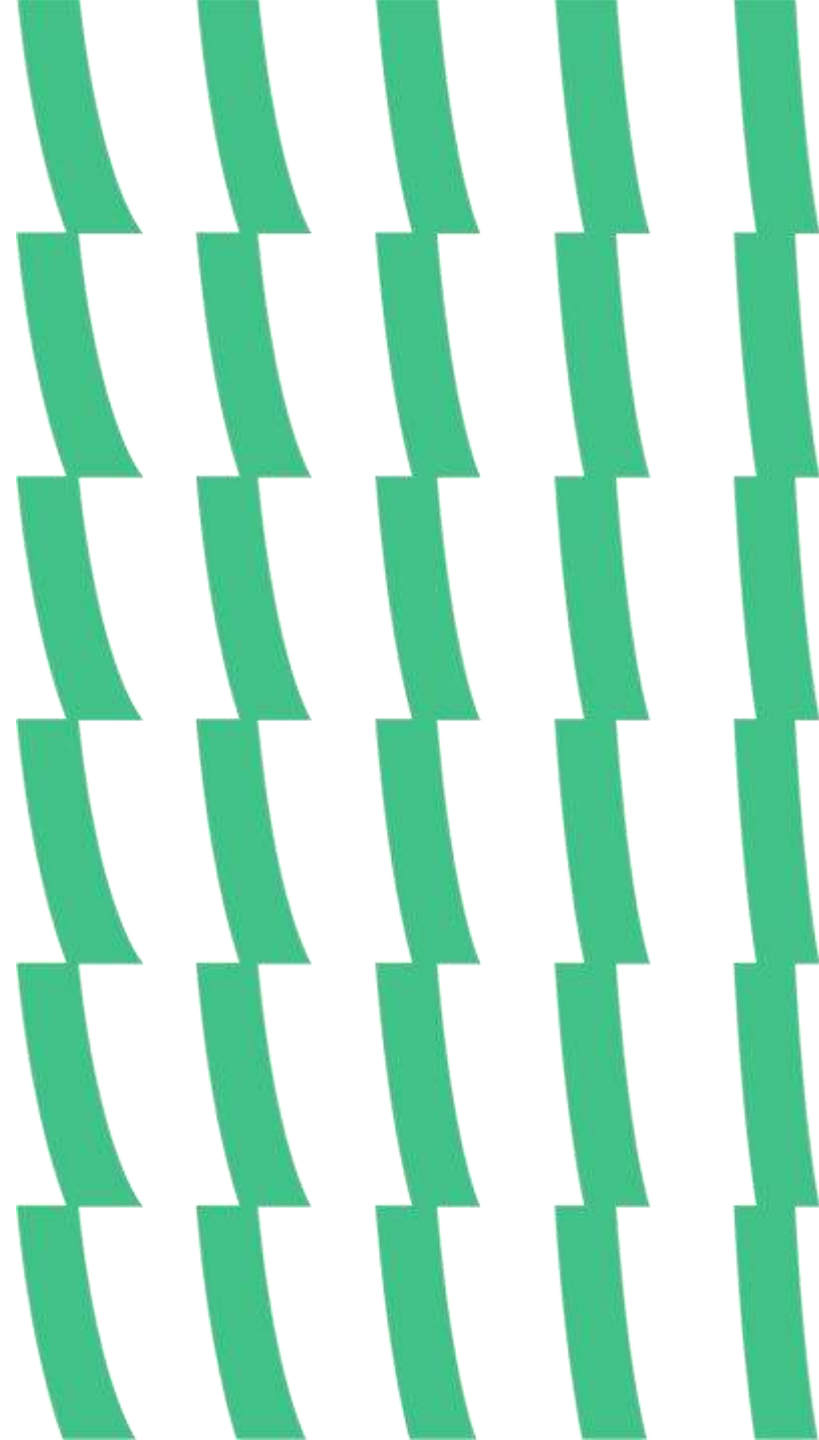
**Тип 0** — “Не изменяй” — заключается в том, что данные после первого попадания в таблицу далее никогда не изменяются. Этот метод практически никем не используется, т.к. он не поддерживает версионности. Он нужен лишь как нулевая точка отсчета для методологии SCD(Slowly Changing Dimensions). По сути, вообще не SCD. Таблица, которая хранит пол - женский, мужской, не определено. Она также не требует ведения истории.

**Тип 1** — “Перезаписывай” — это обычная перезапись старых данных новыми. В чистом виде этот метод тоже не содержит версионности и используется лишь там, где история фактически не нужна

Достоинства: Не добавляется избыточность, Очень простая структура

Недостатки: Не хранит истории :)

Пример: паспортные данные изменились и банально были перезаписаны



# Медленно меняющиеся измерения

## Типы 2

Тип 2 — “Добавляй строку” — для каждой версии создается отдельная запись в таблице с добавлением поля — ключевого атрибута данной версии.

Достоинства:

- Хранит полную и неограниченную историю версий
- Удобный и простой доступ к данным необходимого периода

ID	NAME	NUMBER	TEAM	DATE_START	DATE_END
1	Marc Marquez	93	Honda	11.08.2013	01.01.9999
2	Valentino Rossi	46	Yamaha	11.08.2010	01.01.9999
3	Dani Pedrosa	26	Honda	11.08.2014	01.11.2018
4	Jorge Lorenzo	99	Ducati	01.01.2017	01.01.2019
5	Jorge Lorenzo	99	Honda	02.01.2019	01.01.9999

Недостатки:

- Провоцирует избыточность или заведение дополнительных строк для хранения изменяемых атрибутов измерения;
- Требуеt обновлений в таблице для “закрытия” старых версий.

# Медленно меняющиеся измерения

## Тип 3

**Тип 3** — “Добавляй атрибут” — в самой записи содержатся дополнительные поля для предыдущих значений атрибута. При получении новых данных, старые данные перезаписываются текущими значениями.

Достоинства:

- Небольшой объем данных
- Простой и быстрый доступ к истории

Недостатки:

- Ограниченная история

ID	NAME	NUM	PREVIOUS_TEAM	CURRENT_TEAM	DATE_START
1	Marc Marquez	93	NULL	Honda	11.08.2013
2	Valentino Rossi	46	NULL	Yamaha	11.08.2010
3	Dani Pedrosa	26	NULL	Honda	11.08.2014
4	Jorge Lorenzo	99	Ducati	Honda	02.01.2019



# Медленно меняющиеся измерения

## Тип 4

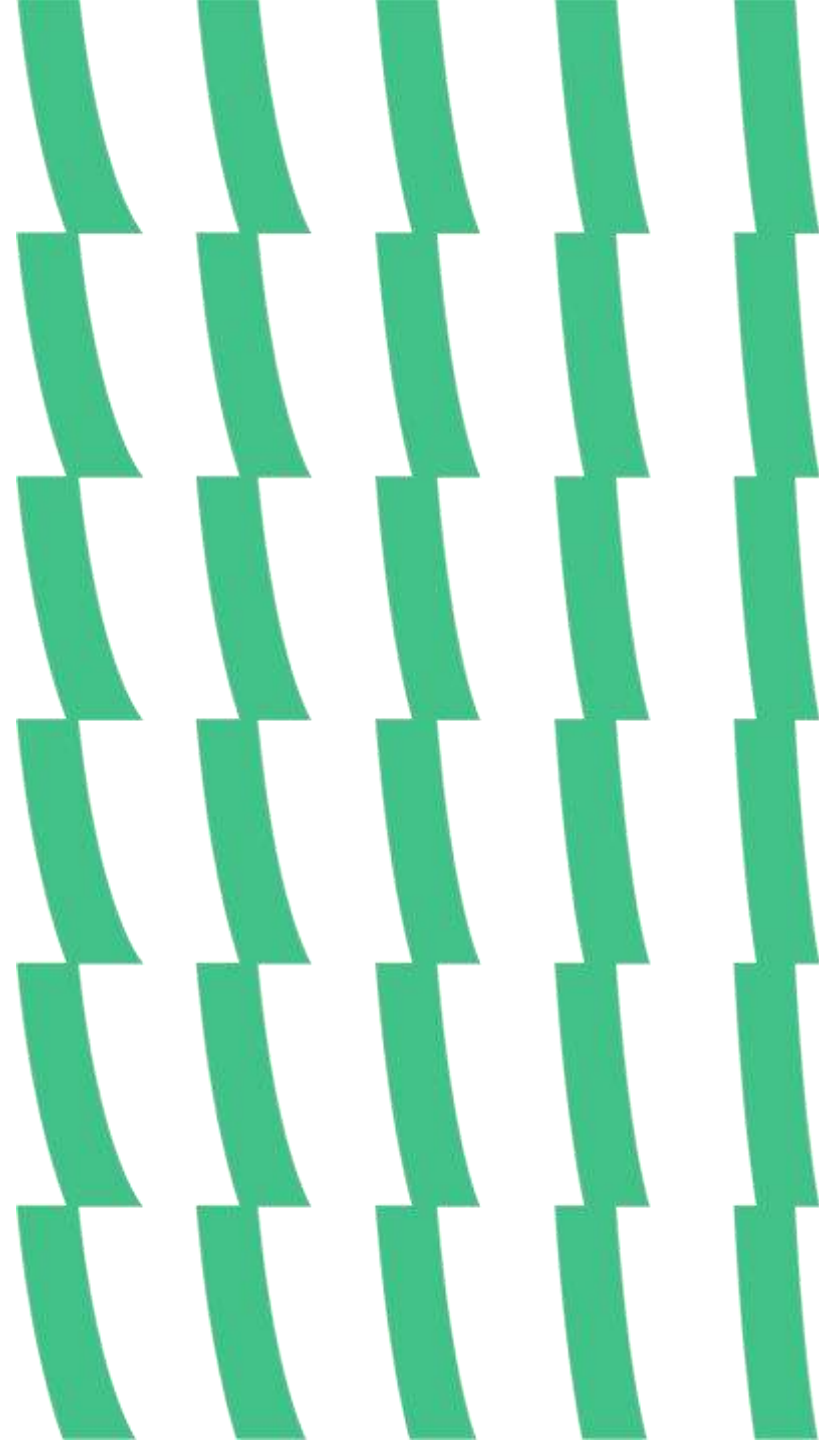
Тип 4 — “Добавляй таблицу” — комбинация SCD1 и SCD2.

1. Первая таблица содержит только текущее состояние данных (SCD1).
2. Вторая таблица содержит полную историю изменения (SCD2).

Данные из первой таблицы регулярно сбрасываются (архивируются) во вторую. Обе таблицы должны содержать одинаковые ключи для повышения производительности запроса.

Достоинства: Быстрая работа с текущими версиями

Недостатки: Разделение единой сущности на разные таблицы



# Медленно меняющиеся измерения

## Тип 6

Название **Тип 6** возникло из идеи комбинации подходов SCD 1, 2 и 3 ( $1+2+3=6$ ).

**Тип 5** при этом пропущен в нумерации и не существует.

ID	VERSION	NAME	NUMBER	TEAM	DATE_START	DATE_END	CURRENT	PREVIOUS_TEAM
1	1	Marc Marquez	93	Honda	11.08.2013	01.01.9999	1	NULL
2	1	Valentino Rossi	46	Yamaha	11.08.2010	01.01.9999	1	NULL
3	1	Dani Pedrosa	26	Honda	11.08.2014	01.11.2018	1	NULL
4	1	Jorge Lorenzo	99	Ducati	01.01.2017	01.01.2019	0	NULL
4	2	Jorge Lorenzo	99	Honda	02.01.2019	01.01.9999	1	Ducati

Тип 6 был придуман Ральфом Кимбаллом как комбинация вышеназванных методов и предназначен для ситуаций, которые они не учитывают или для большего удобства работы с данными. Он заключается во внесении дополнительной избыточности: берется за основу тип 2, добавляется суррогатный атрибут для альтернативного обзора версий (тип 3), и перезаписываются одна или все предыдущие версии (тип 1)

# Быстро меняющиеся измерения

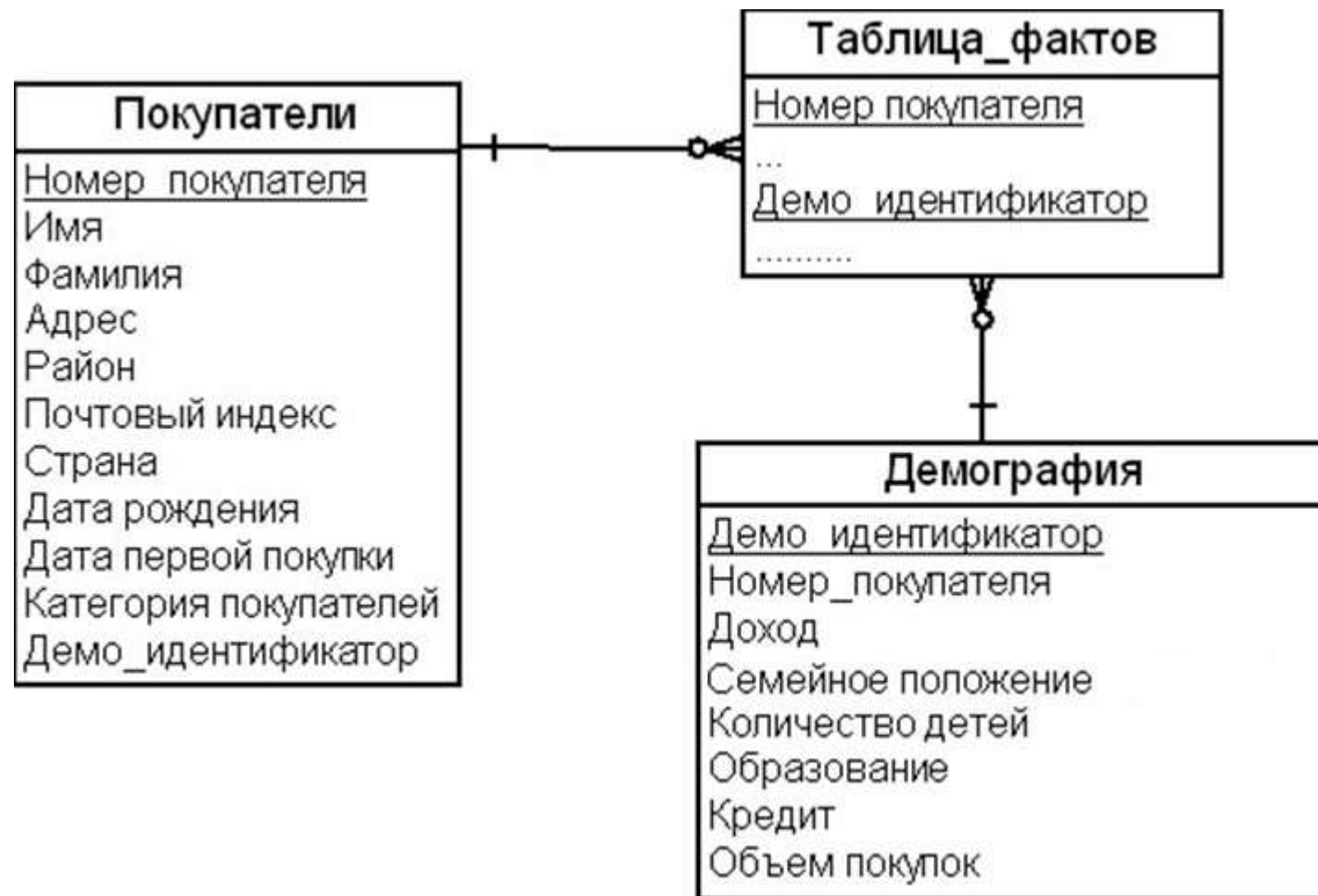
Быстро меняющимися измерениями (Rapidly Changing Dimensions) называются таблицы измерений, в которых некоторые атрибуты могут часто менять свои значения в короткие периоды времени.

Атрибуты таблицы могут меняться с разной скоростью. Если часть полей меняется слишком часто, то их комбинация в пределах одной таблицы приводит к неконтролируемому росту исторических данных.



# Быстро меняющиеся измерения

Для решения проблемы применяется декомпозиция: создаются две сущности, одна для “устойчивых” атрибутов, а другая – для часто меняющихся. Между таблицами-измерениями при этом выстраивается иерархия с использованием внешнего ключа между ними.



# Быстро меняющиеся измерения

Дополнительно можно объединить значения атрибутов по диапазонам. Получаем меньше данных и меньшую детализацию

Ключ	возраст	Количество покупок	Статус
1	20-29	0-25	Обрабатывается
2	30-39	26-50	Текущий
40	50-59	150+	Текущий





# СВЯЗИ

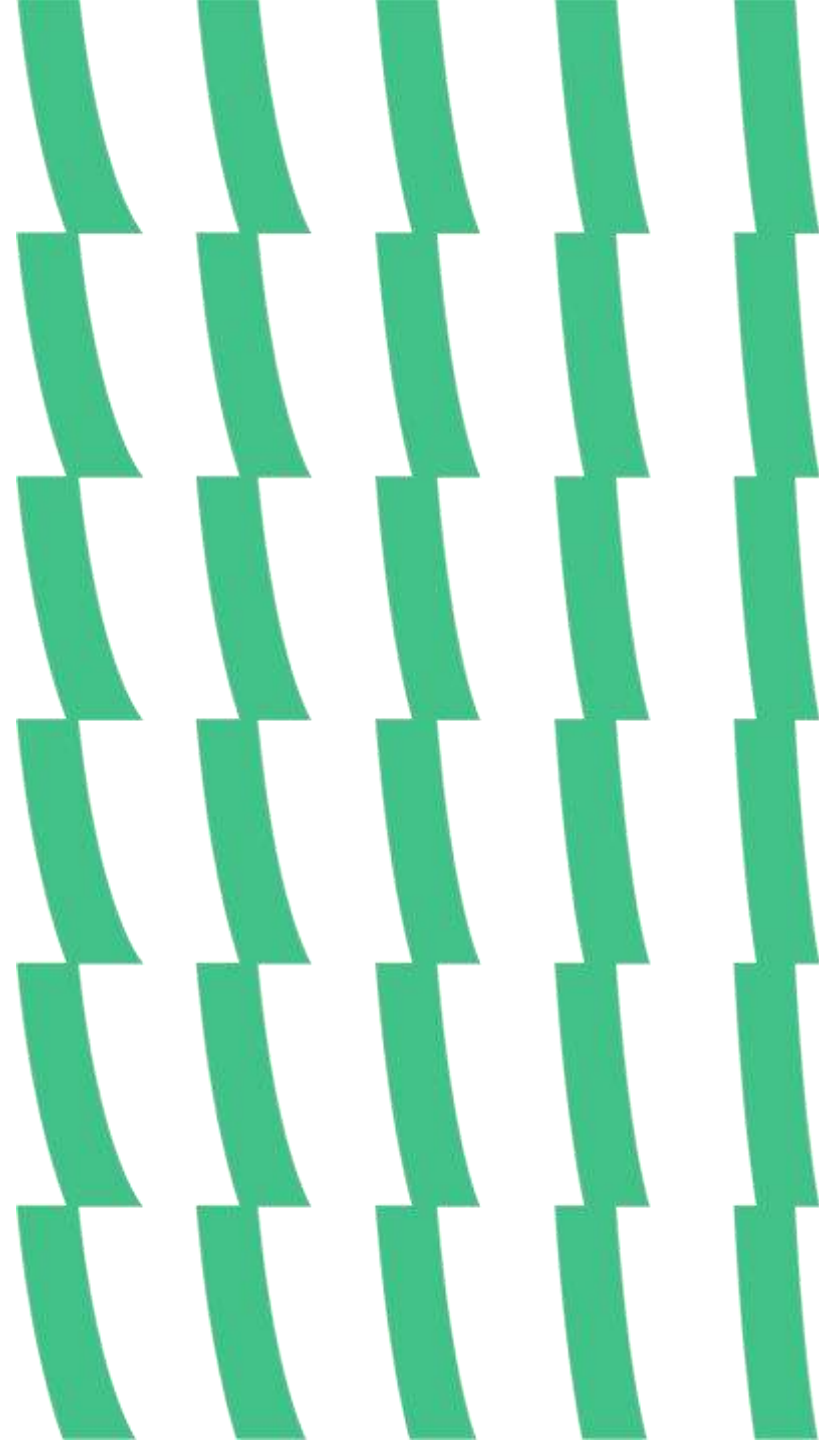
Для организации связи используются внешние ключи. Внешний ключ представляет один или несколько столбцов из одной таблицы, который одновременно является потенциальным ключом из другой таблицы. Внешний ключ необязательно должен соответствовать первичному ключу из главной таблицы. Хотя, как правило, внешний ключ из зависимой таблицы указывает на первичный ключ из главной таблицы.

Связи между таблицами бывают следующих типов:

Один к одному (One to one)

Один к многим (One to many)

Многие ко многим (Many to many)

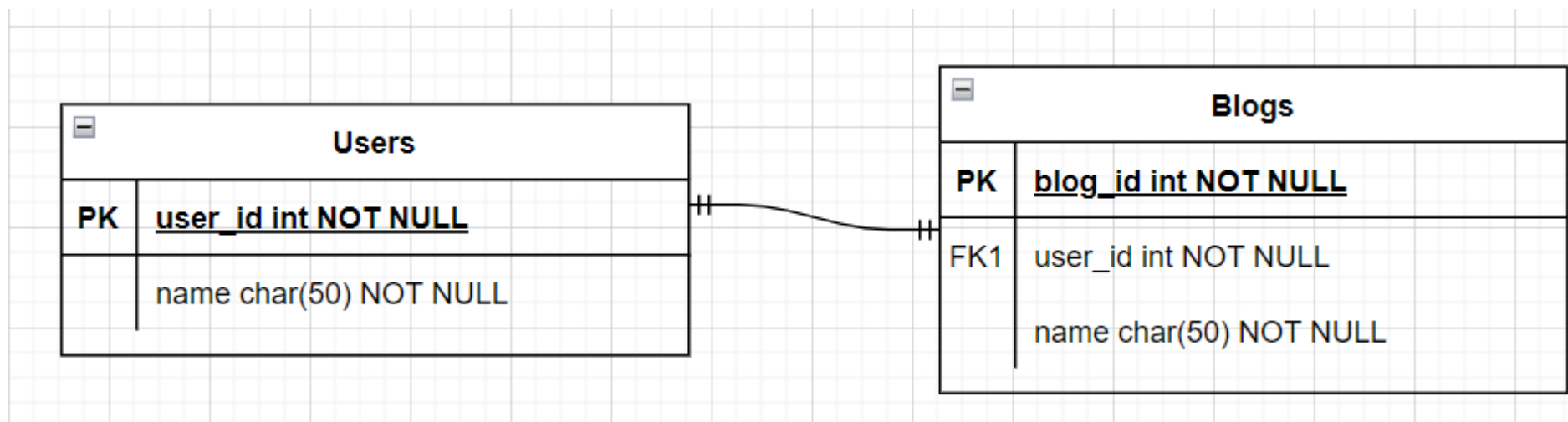


# Один к одному

Данный тип связей встречается не часто. В этом случае объекту одной сущности можно сопоставить только один объект другой сущности. Например, на некоторых сайтах пользователь может иметь только один блог. То есть возникает отношение один пользователь - один блог.

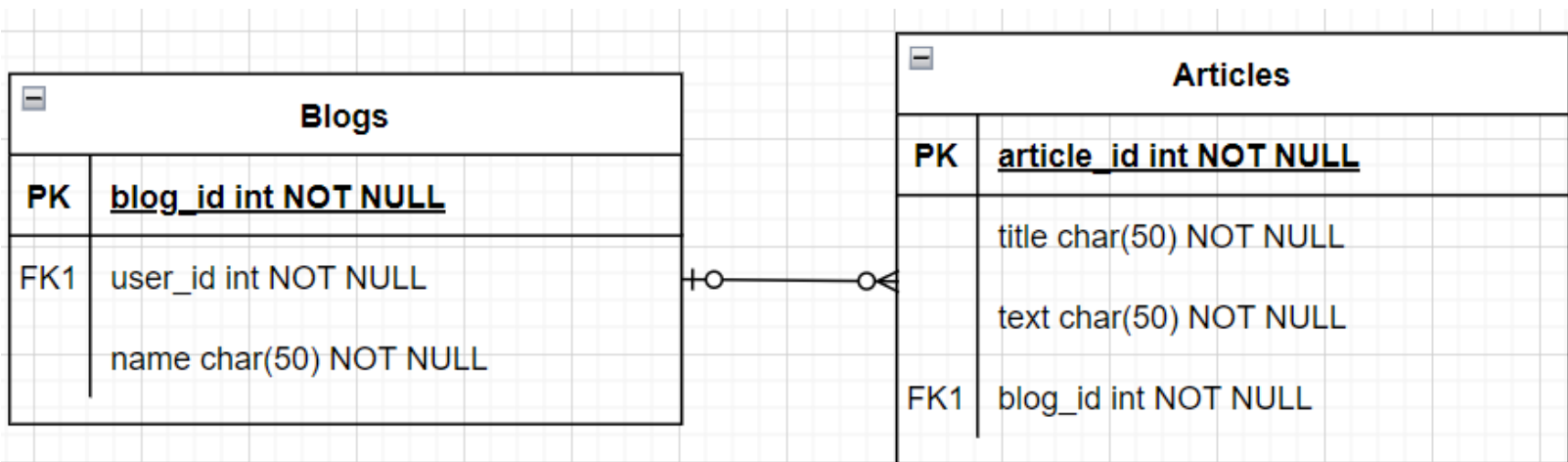
Нередко этот тип связей предполагает разбиение одной большой таблицы на несколько маленьких. Основная родительская таблица в этом случае продолжает содержать часто используемые данные, а дочерняя зависимая таблица обычно хранит данные, которые используются реже.

В этом отношении первичный ключ зависимой таблицы в то же время является внешним ключом, который ссылается на первичный ключ из главной таблицы.



# Один ко многим

Это наиболее часто встречаемый тип связей. В этом типе связей несколько строк из дочерней таблицы зависят от одной строки в родительской таблице. Например, в одном блоге может быть несколько статей. В этом случае таблица блогов является родительской, а таблица статей - дочерней. То есть один блог - много статей. Или другой пример, в футбольной команде может играть несколько футболистов. И в то же время один футболист одновременно может играть только в одной команде. То есть одна команда - много футболистов.

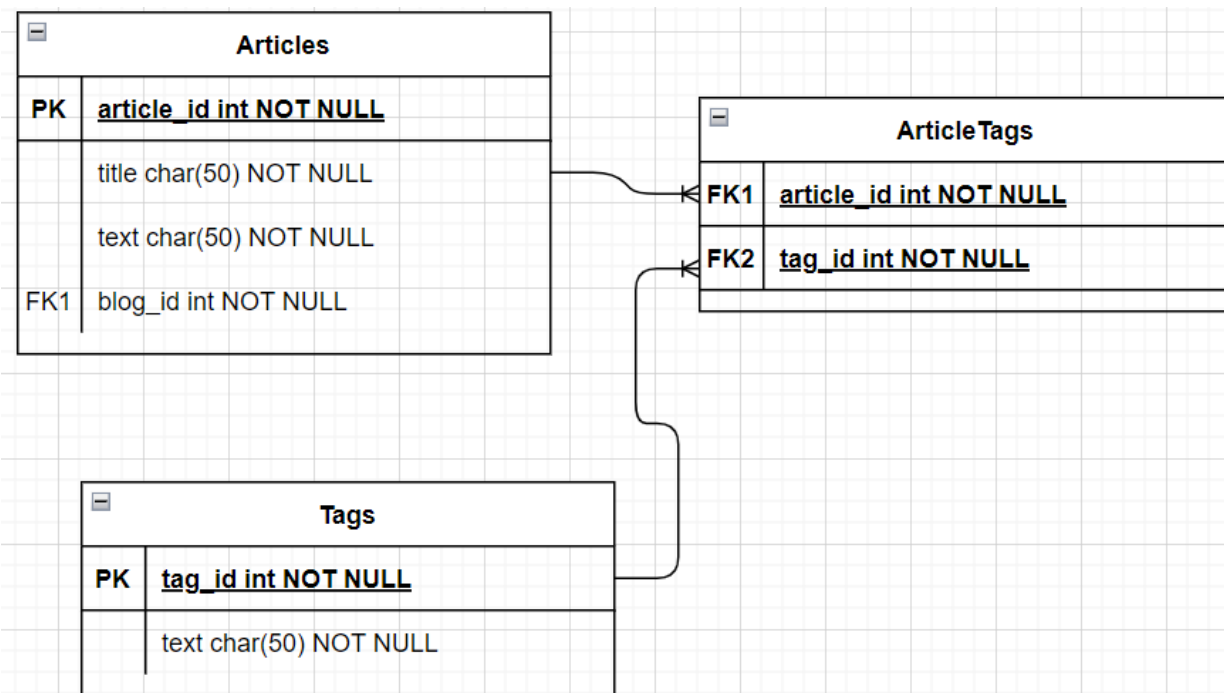


# Многие ко многим

При этом типе связей одна строка из таблицы А может быть связана с множеством строк из таблицы В. В свою очередь одна строка из таблицы В может быть связана с множеством строк из таблицы А. Типичный пример - студенты и курсы: один студент может посещать несколько курсов, и соответственно на один курс могут записаться несколько студентов.

Другой пример - статьи и теги: для одной статьи можно определить несколько тегов, а один тег может быть определен для нескольких статей.

Это делается посредством вспомогательной промежуточной таблицы. Иногда данные из этой промежуточной таблицы представляют отдельную сущность.



Спасибо  
за внимание!