DWH

Лекция №2

Подходы к построению Хранилищ данных Билла Инмона и Ральфа Кимбалла

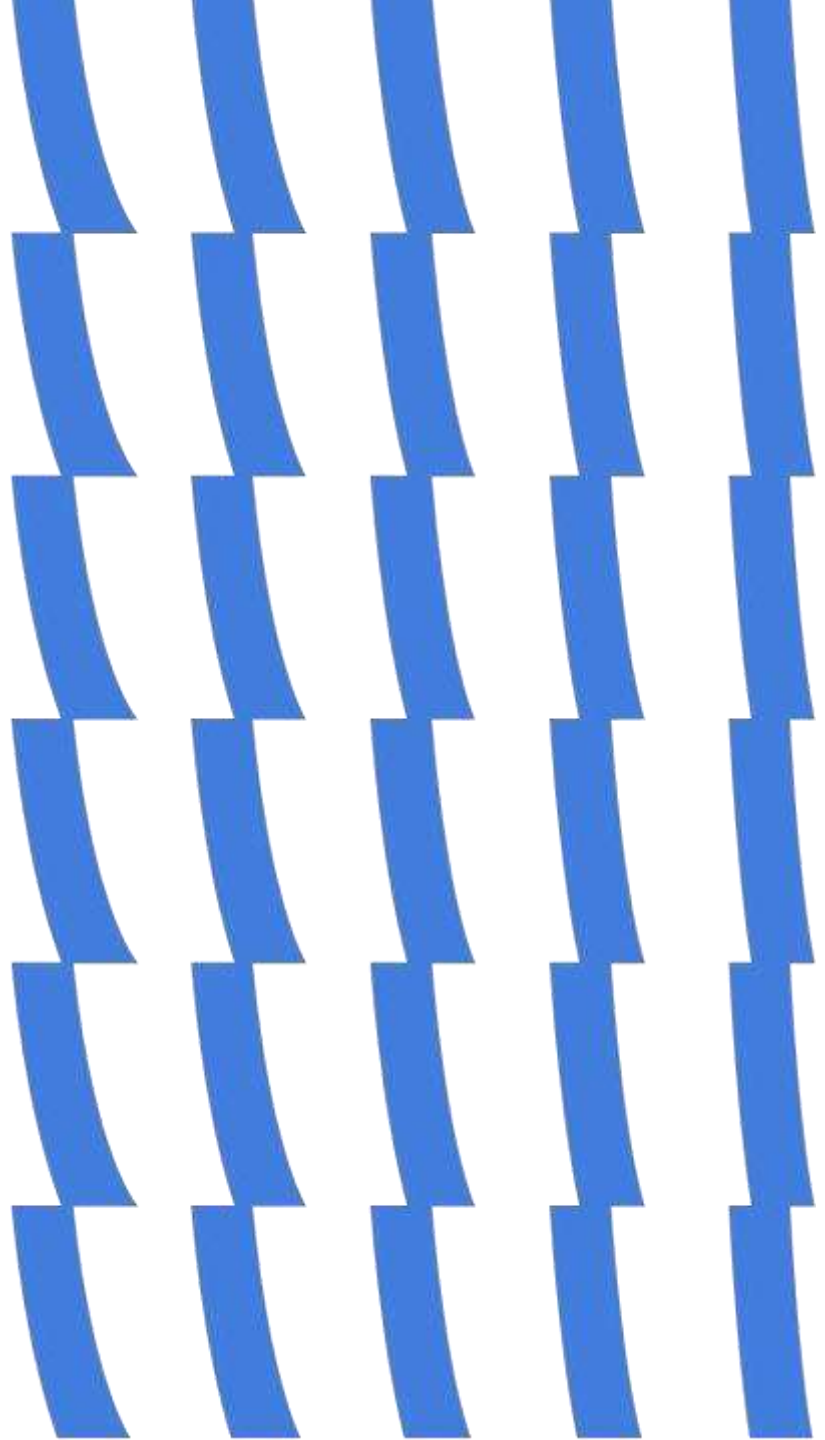
В прошлой лекции

Реляционные БД

Данные и связи между данными хранятся в таблицах(отношения). Каждая строка – отдельная запись(кортеж). Каждый столбец имеет своё имя и тип данных.

ACID — набор требований к транзакционным системам, обеспечивающий наиболее надёжную и предсказуемую её работу

Атомарность Согласованность Изолированность Стойкость

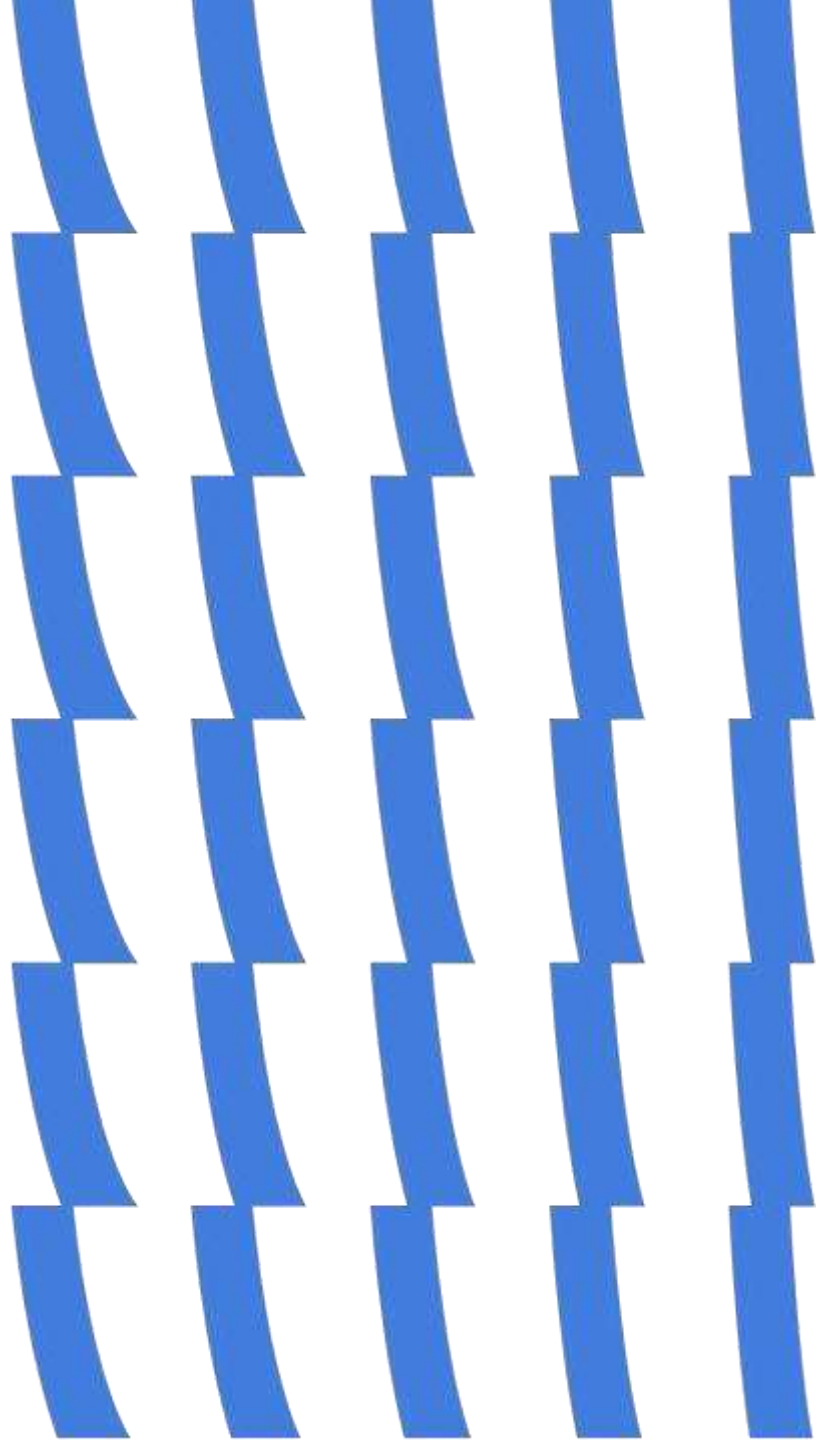


NoSQL

NoSQL решения способны обеспечивать горизонтальную масштабируемость и большую отзывчивость, но жертвуют при этом надежностью транзакций. NoSQL работает с документами без схемы, которые хранят данные в документах, графе, ключе-значении и неупорядоченном виде.

Base вместо ACID:

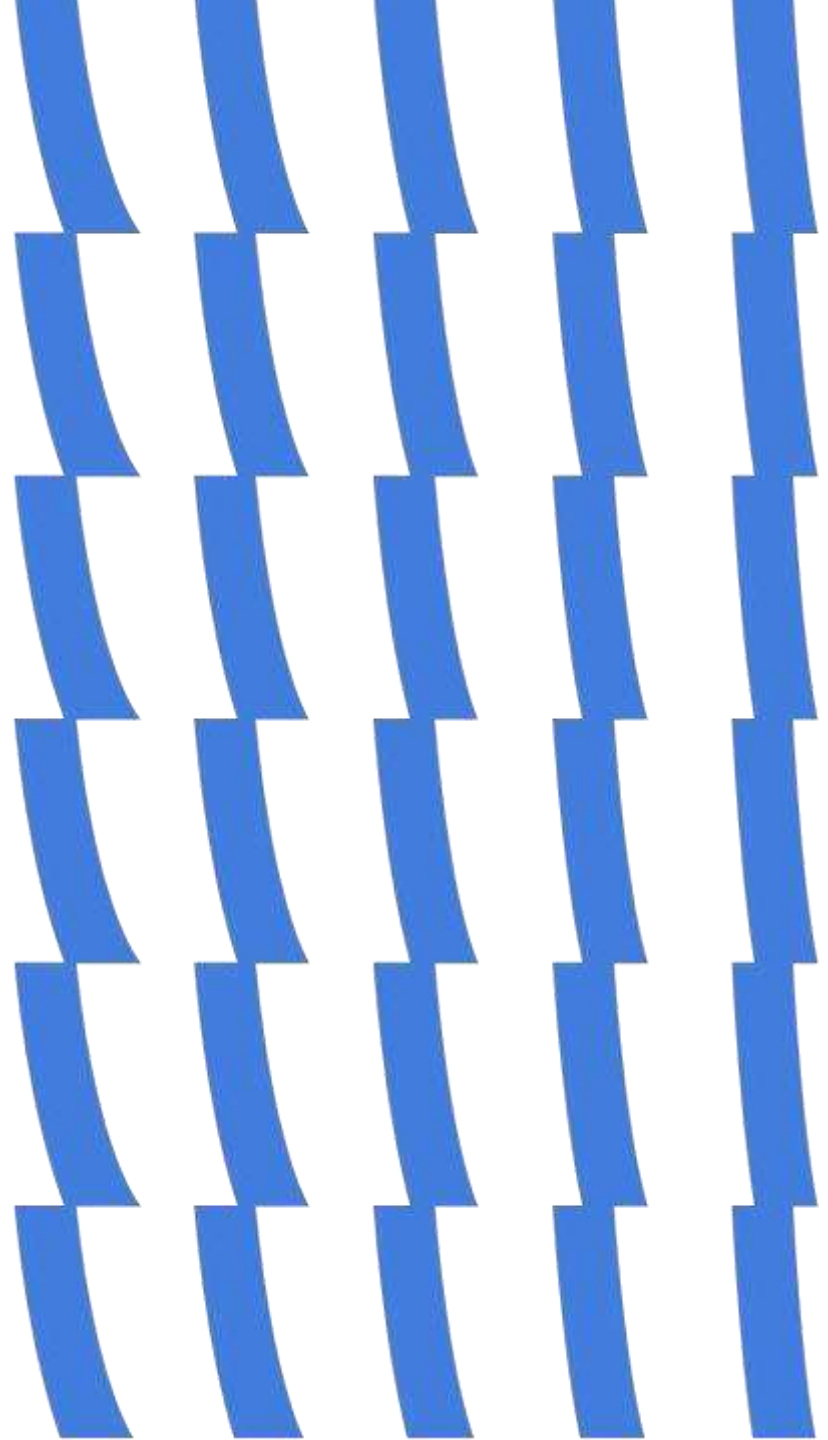
- Basic Availability — базовая доступность.
- Soft state — гибкое состояние.
- Eventual consistency – согласованность в конечном счете



CAP-теорема

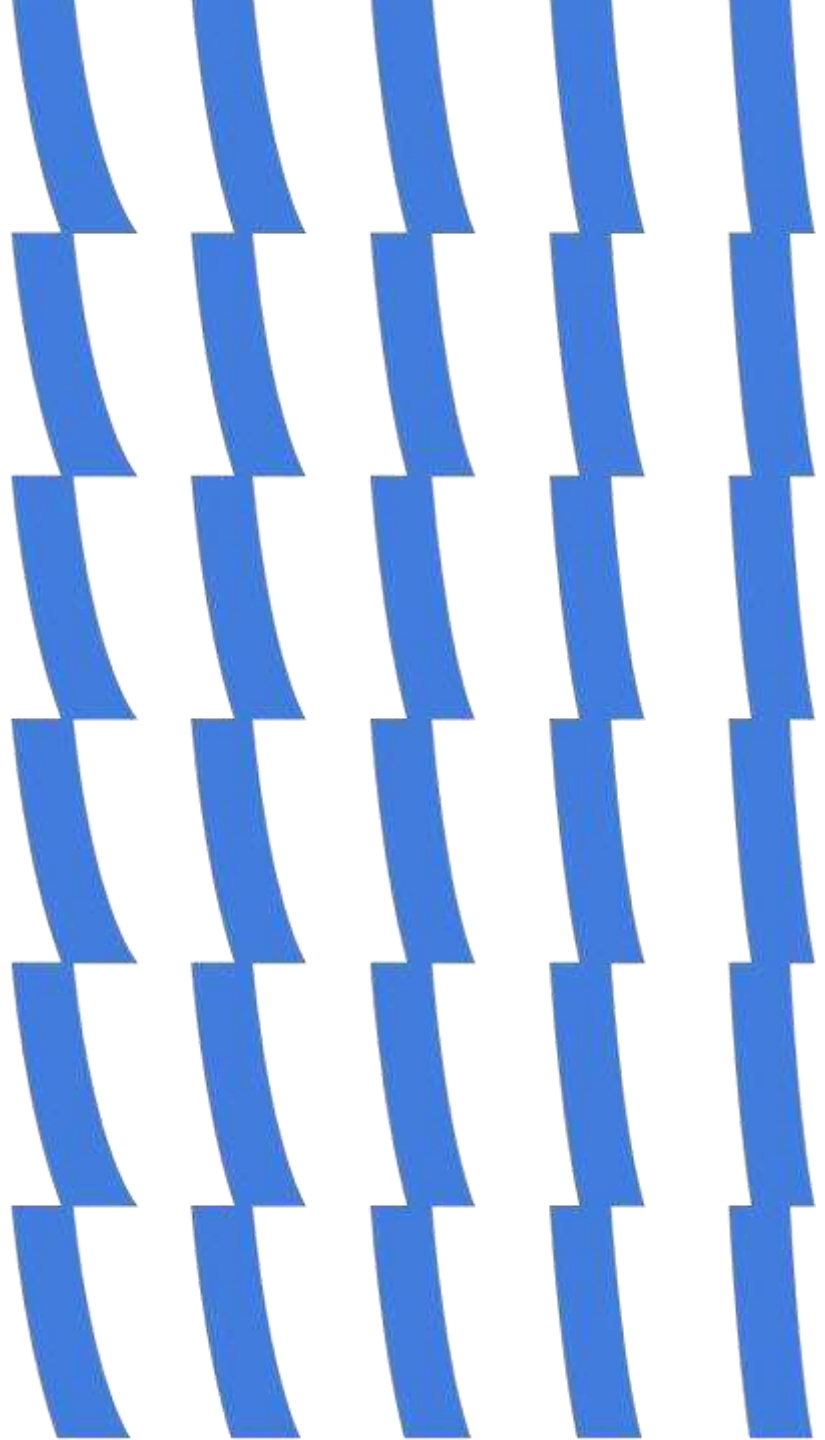
CAP-теорема, возможно достичь только два свойства

- C (consistency) — Согласованность
- A (availability) — Доступность
- P (partition tolerance) — Устойчивость к распределению.



NewSQL

Концепция NewSQL сочетает в себе масштабируемость NoSQL подхода и полную поддержку ACID-транзакционности классических реляционных баз.



Большие данные 5V

1

Объём (volume, в смысле величины физического объёма)

2

Скорость (velocity в смыслах как скорости прироста, так и необходимости высокоскоростной обработки и получения результатов)

3

Многообразие (variety, в смысле возможности одновременной обработки различных типов структурированных и полуструктурированных данных)

4

veracity — Достоверность. Целостность данных и возможность доверять полученным на их основе результатам.

5

value — Ценность / экономическая целесообразность обработки соответствующие объёмов в конкретных условиях.

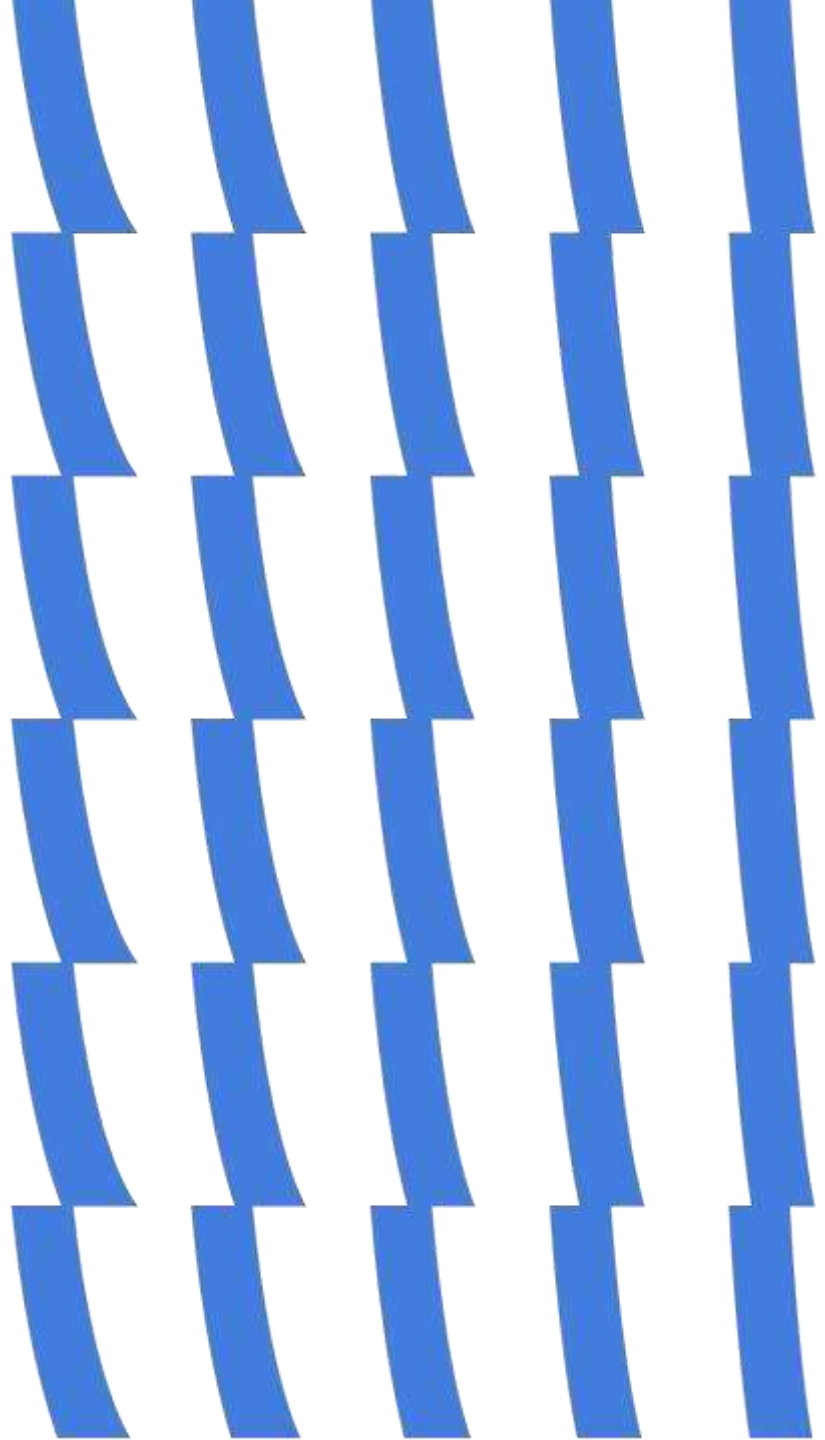
OLTP vs. OLAP

OLTP (Online Transaction Processing) обработка транзакций в режиме реального времени используется для операционной работы конкретной системы. OLTP-система характеризуется большим потоком коротких транзакций (INSERT, UPDATE, DELETE). Ключевые особенности OLTP-систем: почти мгновенная обработка запросов, обеспечение целостности данных в средах с множественным доступом и высокая нагруженность (определяется числом транзакций в единицу времени)

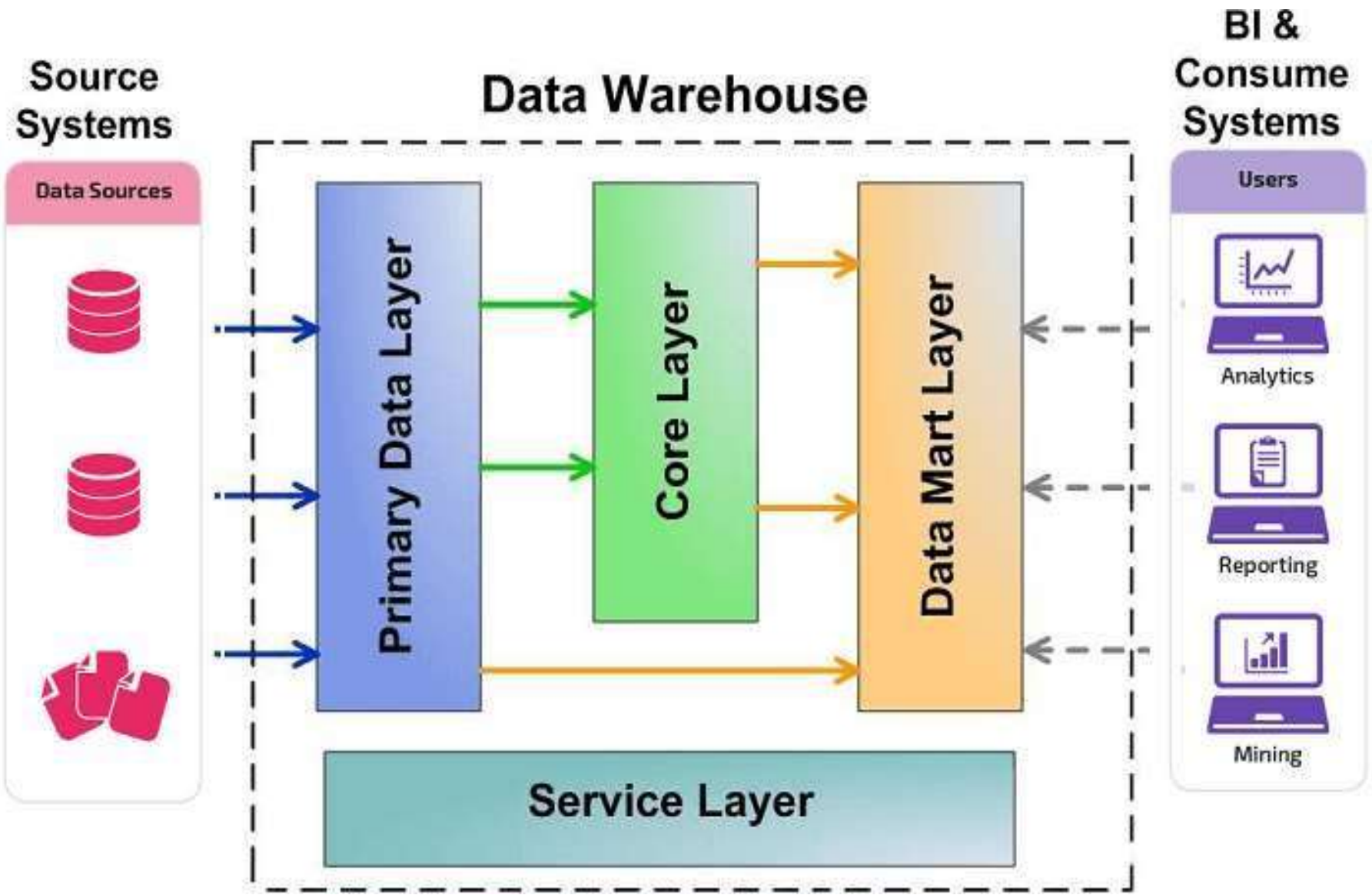
OLAP (Online Analytical Processing) интерактивная аналитическая обработка имеет дело с историческими (архивными) данными. Такая система характеризуется относительно низкой частотой транзакций, но большими объёмами затрагиваемых данных. Запросы часто очень сложны и включают агрегирование (группировки). Для OLAP-систем показатель эффективности — это время отклика, а данные в OLAP-базе разложены в многомерную схему под конкретный сценарий работы с ними (например, “звезда”)

Классическое определение DWH (Data Warehouse)

Хранилище данных — предметно-ориентированный, интегрированный, некорректируемый, поддерживающий хронологию набор данных, организованный для целей поддержки принятия решений.



Слои данных в DWH



Реляционная модель

Модель данных СУБД

Модель данных СУБД — формальная теория представления и обработки данных в системе управления базами данных (СУБД), которая включает, по меньшей мере, три аспекта:

- Аспект структуры — что из себя логически представляет база данных.
- Аспект целостности — определяет средства описаний корректных состояний базы данных.
- Аспект манипулирования — определяет способы перехода между состояниями БД (то есть способы модификации данных) и способы извлечения данных из БД.

Модель данных описывает абстрактную машину доступа к данным, с которой взаимодействует пользователь. То есть речь идёт о логической модели, а не способе физического хранения объектов и структур.

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.

Реляционная алгебра

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных. Результат каждой из операций также является отношением.

Первоначальный набор из 8 операций был предложен Э. Коддом в 1970-е годы и включал как операции, которые до сих пор используются (проекция, соединение и т.д.), так и операции, которые не вошли в употребление (например, деление отношений).

Поскольку многие операции выразимы друг через друга, в составе реляционной алгебры можно выделить несколько вариантов базиса (минимальный набора операций, через которые можно выразить все остальные). Наиболее известный и строго определённый базис предложен Кристофером Дейтом и Хью Дарвеном, который использует 5 операторов: and, or, not, remove, rename.

Основные понятия

Атрибут — свойство некоторой сущности. Часто называется полем таблицы.

Домен атрибута — множество допустимых значений, которые может принимать атрибут.

Кортеж — конечное множество взаимосвязанных допустимых значений атрибутов, которые вместе описывают некоторую сущность (строка таблицы).

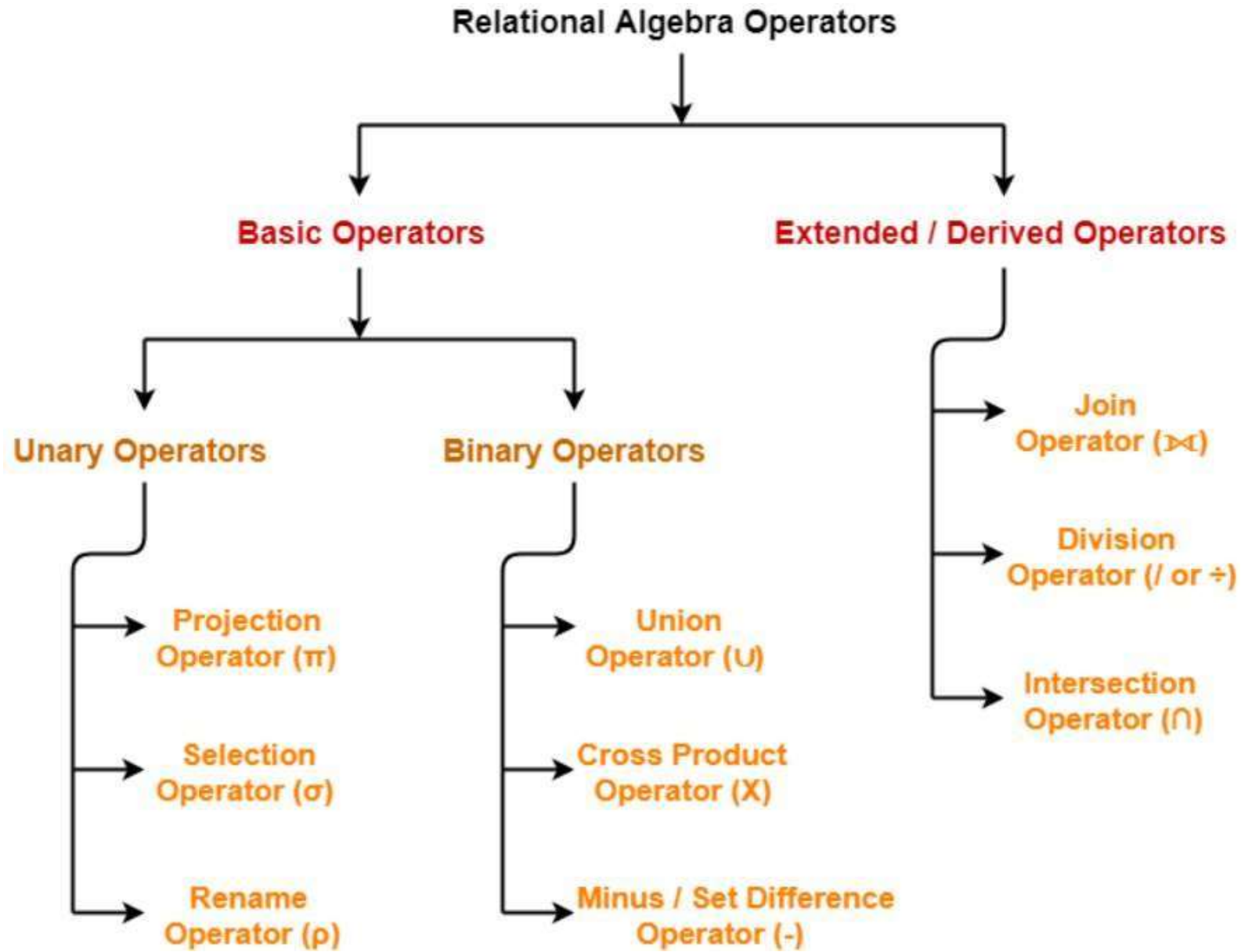
Отношение — конечное множество кортежей (таблица).

Соответствие основных понятий

Термин реляционной алгебры	Термин реляционных БД
База данных	Набор таблиц
Схема базы данных	Набор заголовков таблиц
Отношение	Таблица
Заголовок отношения	Заголовок таблицы
Тело отношения	Тело таблицы
Атрибут отношения	Наименование столбца таблицы
Кортеж отношения	Строка таблицы
Степень (-арность) отношения	Количество столбцов таблицы
Мощность отношения	Количество строк таблицы
Домены и типы данных	Типы данных в ячейках таблицы

Основные понятия

Основные операции



Основные операции

Поставщики

PNUM	Budget	PNAME
1	10M	Фирма 1
2	12M	Фирма 2
4	7M	Фирма 4
3	5M	Фирма 3

Детали

PNUM	DNUM	VOLUME
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	3	1000

Division operator

Пусть имеются отношения A(X, Y) и B (Y), где атрибуты Y определены на одном и том же домене. Тогда результатом деления A ÷ B будет отношение с заголовком из атрибута X и телом, в которое входят кортежи такие, что существует кортеж , который принадлежит отношению A для всех кортежей из отношения B.

SP

SID	PID
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S5	P5

÷

P

PID
P1

=

SID
S1
S2

SP

SID	PID
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S5	P5

÷

P

PID
P2
P4

=

SID
S1
S4

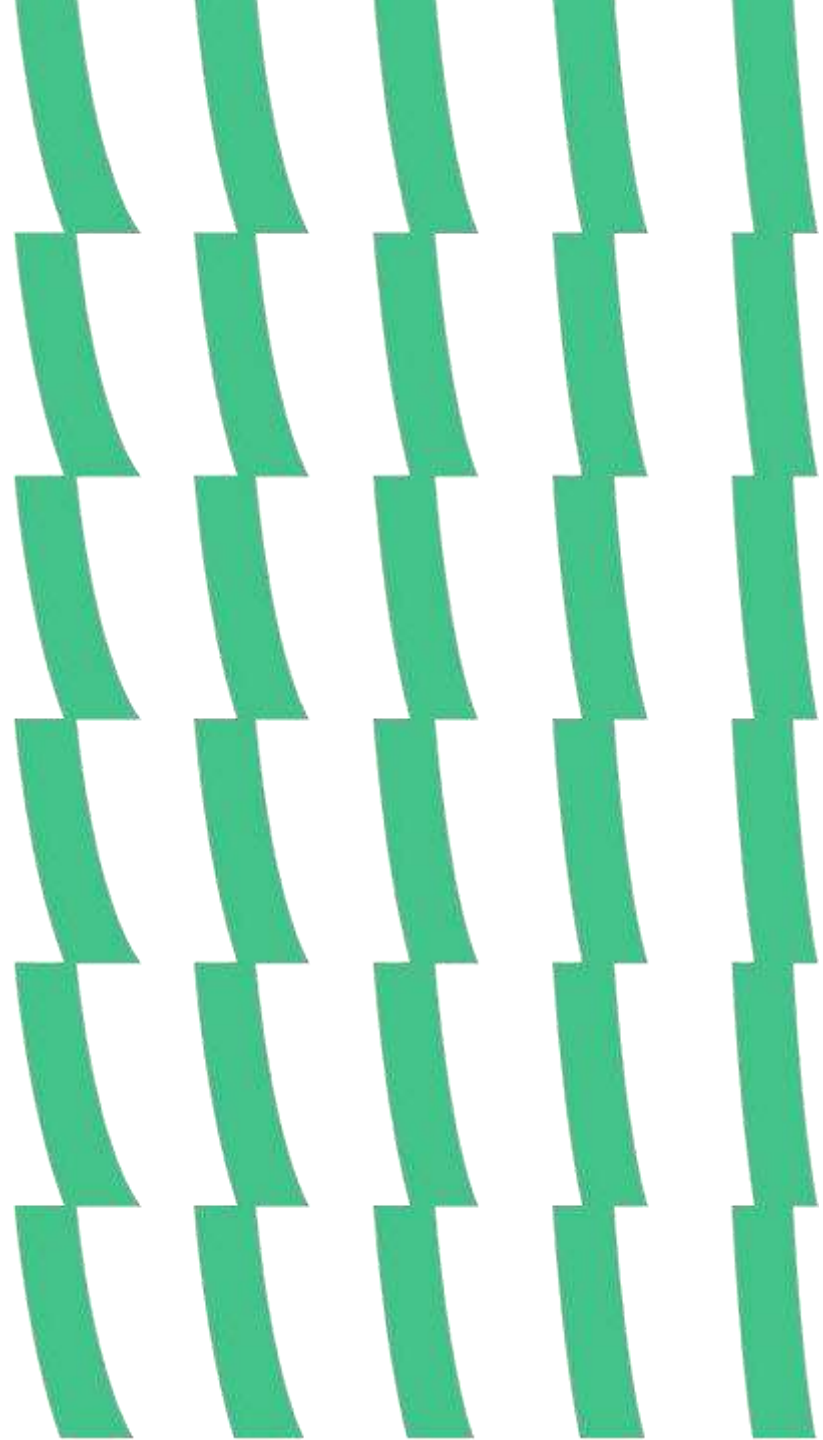
Свойства отношения

- В отношении нет одинаковых кортежей. Тело отношения есть множество кортежей и, как всякое множество, не может содержать неразличимые элементы.
- Порядок кортежей не определён. Причина та же — тело отношения есть множество, а множество не упорядочено. Одно и то же отношение может быть изображено разными таблицами, в которых строки идут в различном порядке.
- Порядок атрибутов не определён. Т. к. каждый атрибут имеет уникальное имя в пределах отношения, то порядок атрибутов не имеет значения. Одно и то же отношение может быть изображено разными таблицами, в которых столбцы идут в различном порядке.

Отношение vs. Таблица

Вообще говоря, в таблице могут не выполняться некоторые свойства отношения, то есть не всякая таблица есть корректное и полное представление некоторого отношения.

Поэтому для применения реляционной модели на практике вводят понятие так называемых нормальных форм для отношений.



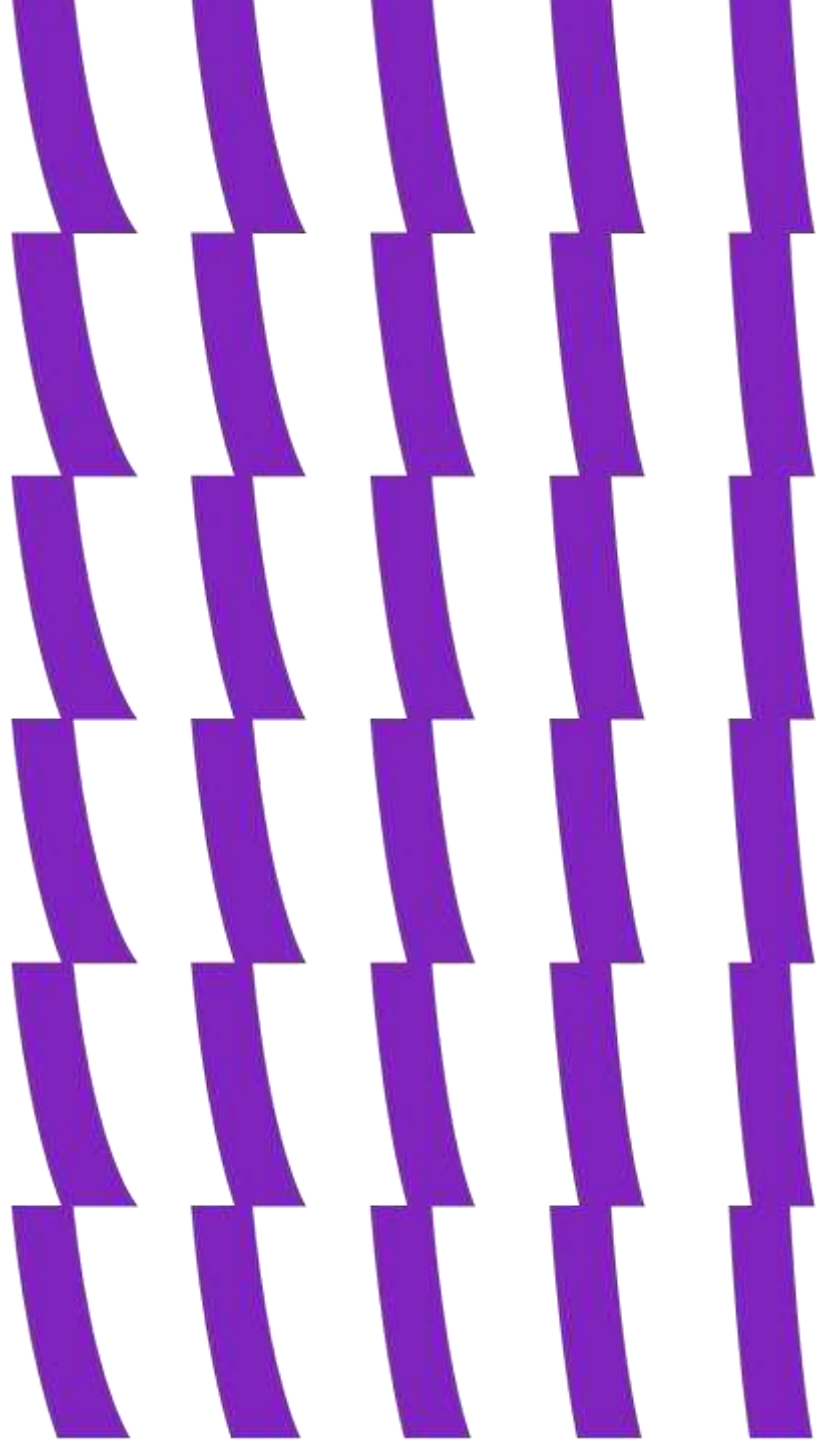
Нормализация

1НФ

Отношение находится в первой нормальной форме (сокращённо 1НФ), если все его атрибуты атомарны, то есть ни один из его атрибутов нельзя разделить на более простые атрибуты, которые соответствуют каким-то другим свойствам описываемой сущности.

В реляционной модели отношение всегда находится в первой нормальной форме по определению. Однако для таблиц это может быть не верно.

Атомарность атрибута — плохо формализуемое понятие и обычно зависит от контекста, в котором используются данные таблицы.



1НФ

Согласно Дейту таблица находится в 1НФ тогда и только тогда, когда она является прямым и верным представлением некоторого отношения. Для этого должно выполняться 5 свойств:

- Порядок строк не несёт дополнительной информации.
- Порядок столбцов не несёт дополнительной информации.
- Нет повторяющихся строк.
- Каждая ячейка таблицы содержит ровно одно значение из соответствующего домена и ничего больше.
- Все колонки “регулярны”, т. е. не содержат скрытых компонент со ссылками на посторонние объекты (адрес строки, указатель на внешний объект, скрытая временная метка и др.).

1НФ

Для того чтобы нормализовать исходное отношение, атрибуты которого неатомарны, необходимо объединить схемы основного и подчинённого отношений. Задача может усложняться тем, что значение неатомарного атрибута может содержать несколько кортежей.

Id_сотруд	Фамилия	Должность	Проекты
1	Иванов	Программист	ID: 123; Название: Система управления паровым котлом; Дата сдачи: 30.09.2011 ID: 231; Название: ПС для контроля и оповещения о превышениях ПДК различных газов в помещении; Дата сдачи: 30.11.2011 ID: 321; Название: Модуль распознавания лиц для защитной системы; Дата сдачи: 01.12.2011

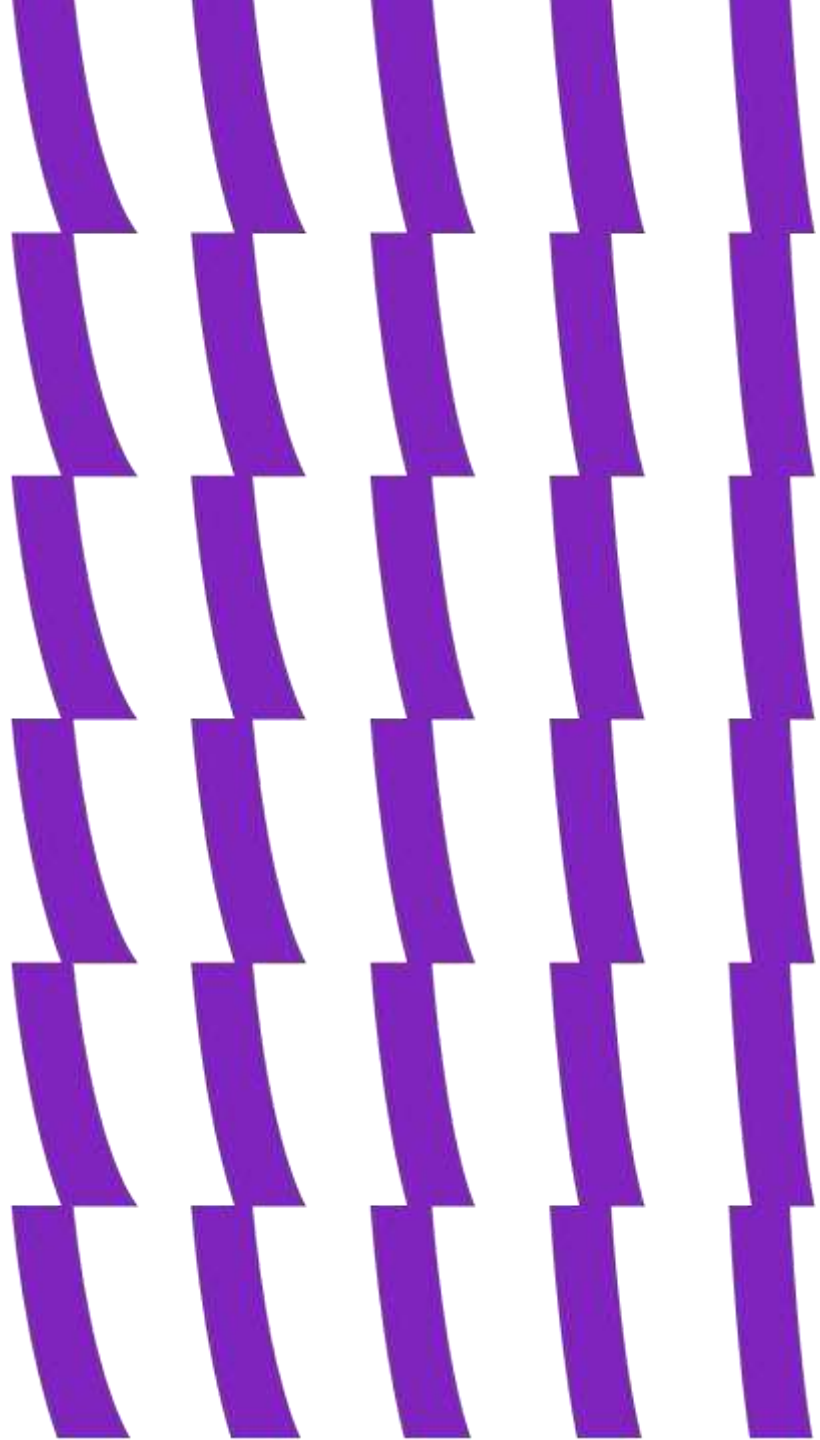
1НФ

Id_сотруд	Фамилия	Должность	Id_проекта	Название	Дата сдачи
1	Иванов	Программист	123	Система управления паровым котлом	30.09.2011
1	Иванов	Программист	231	ПС для контроля и оповещения о превышениях ПДК различных газов в помещении	30.11.2011
1	Иванов	Программист	321	Модуль распознавания лиц для защитной системы	01.12.2011

2НФ

Дадим формальное определение, которое сейчас никто не поймёт ;)

Отношение находится во второй нормальной форме (сокращённо 2НФ) тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от (каждого) её потенциального ключа.



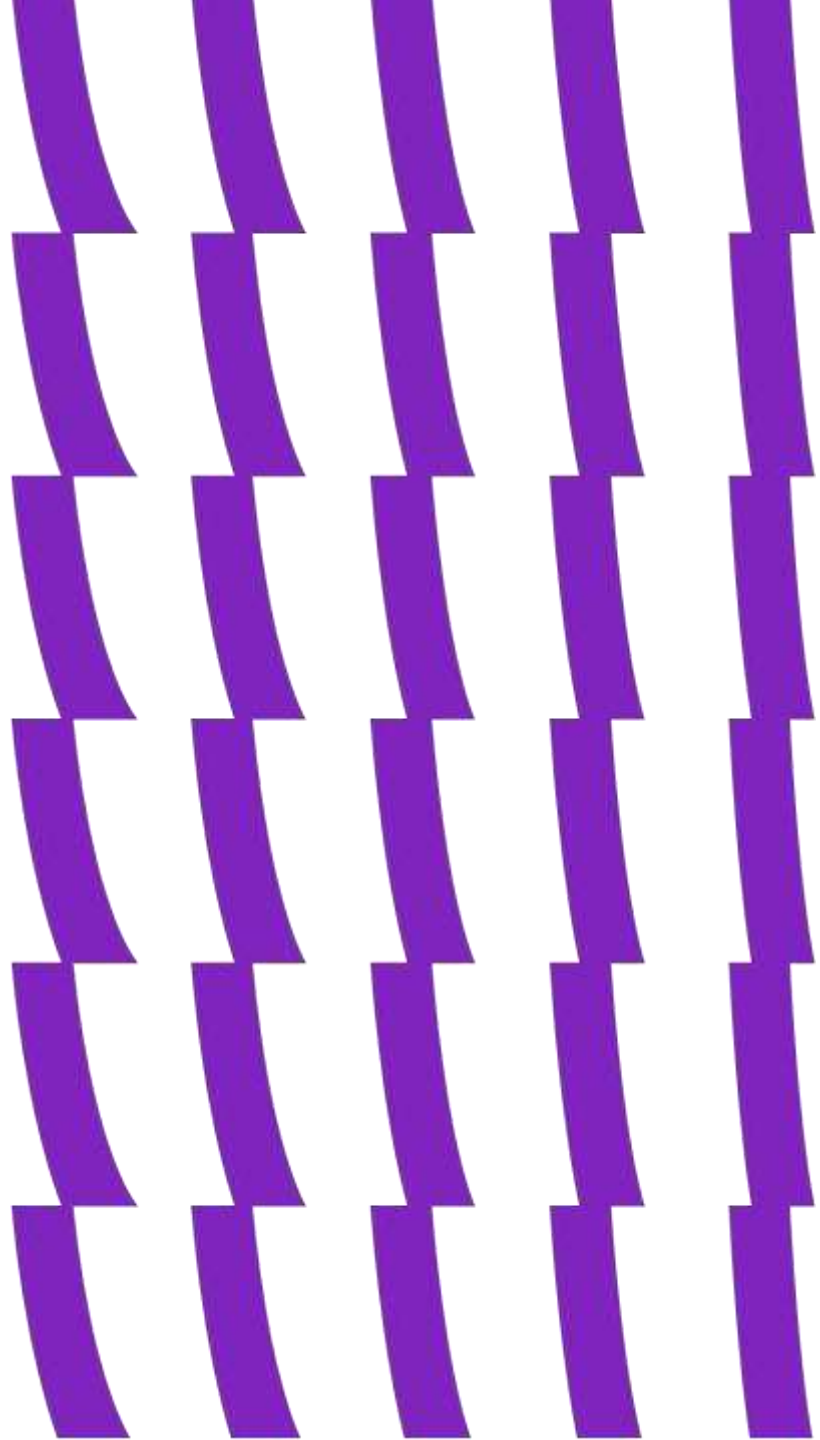
Потенциальный ключ

Подмножество атрибутов отношения будем называть потенциальным ключом, если оно обладает следующими свойствами:

- Уникальность — в отношении не может быть двух различных кортежей, с одинаковым значением выбранных атрибутов.
- Неизбыточность — никакое подмножество атрибутов потенциального ключа не обладает свойством уникальности.

Потенциальный ключ, состоящий из одного атрибута, называется простым. Потенциальный ключ, состоящий из нескольких атрибутов, называется составным.

Отношение может иметь несколько потенциальных ключей. Традиционно один из потенциальных ключей объявляется первичным, а остальные — альтернативными.

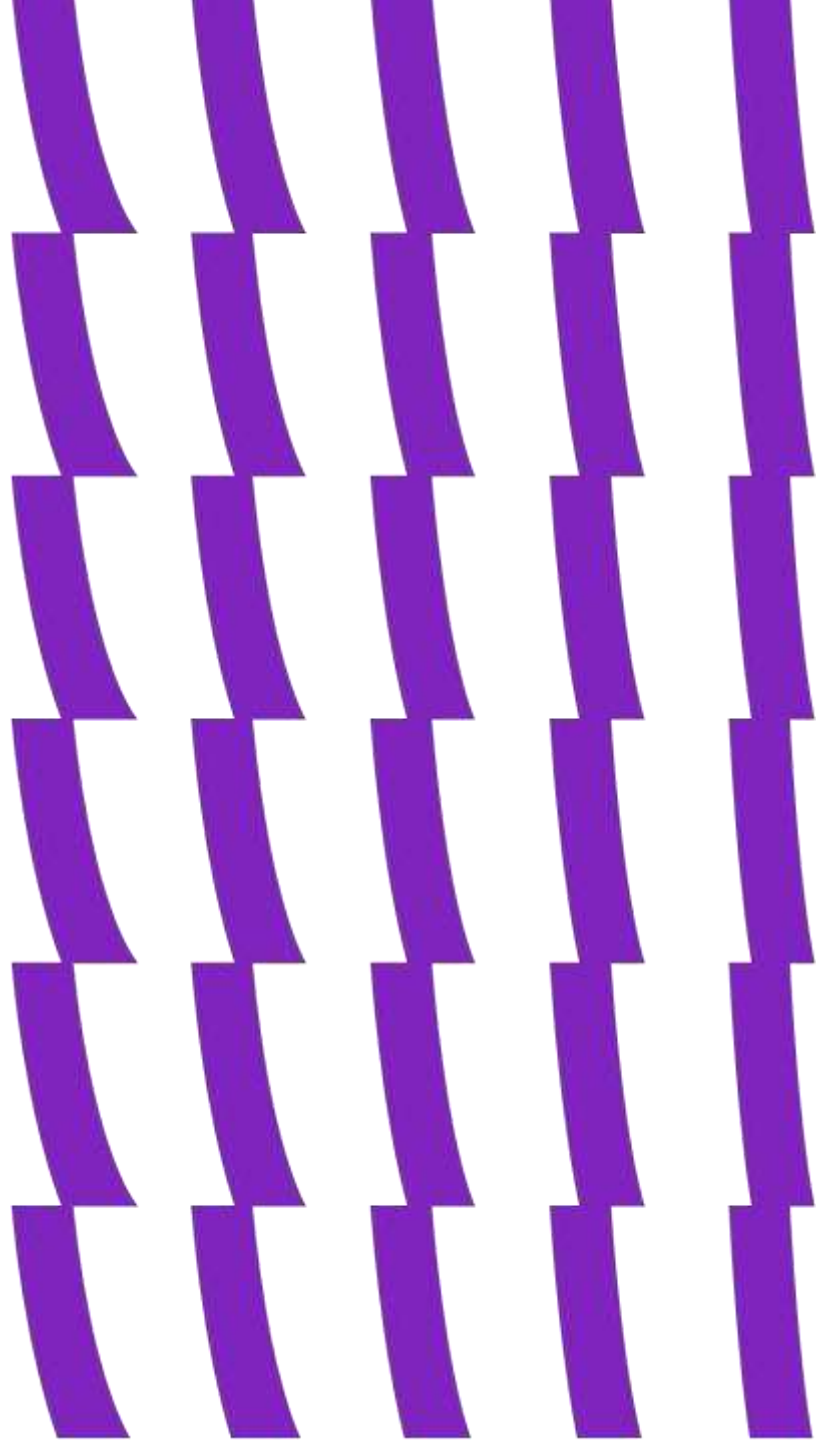


Функциональная зависимость

Функциональная зависимость между атрибутами (множествами атрибутов) X и Y означает, что для любого допустимого набора кортежей в данном отношении: если два кортежа совпадают по значению X, то они совпадают по значению Y.

Например, если значение атрибута «Название компании» — «VK», то значением атрибута «Адрес» в таком кортеже всегда будет «Россия, 125167, г.Москва,Ленинградский пр. д.39, стр.79»

Обозначение: {X} -> {Y}.



2НФ

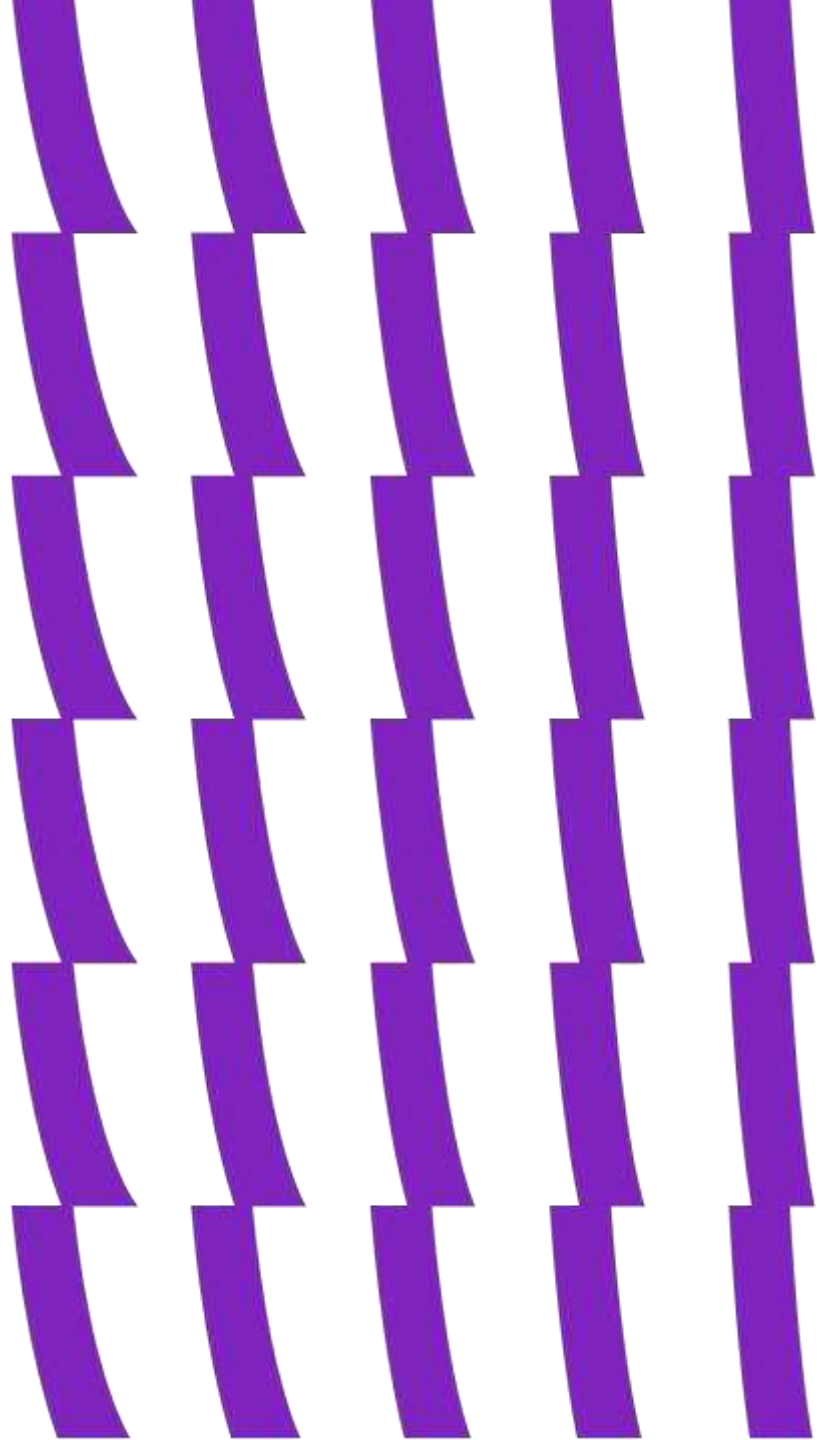
Повторим определение!

Отношение находится во второй нормальной форме тогда и только тогда, когда оно находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от (каждого) её потенциального ключа.

А теперь чуть более человеческим языком:

Нет функциональной зависимости от части ключа к неключевому атрибуту.

Неприводимость означает, что функциональную зависимость “ключ —> неключевой атрибут” нельзя редуцировать до подключа.



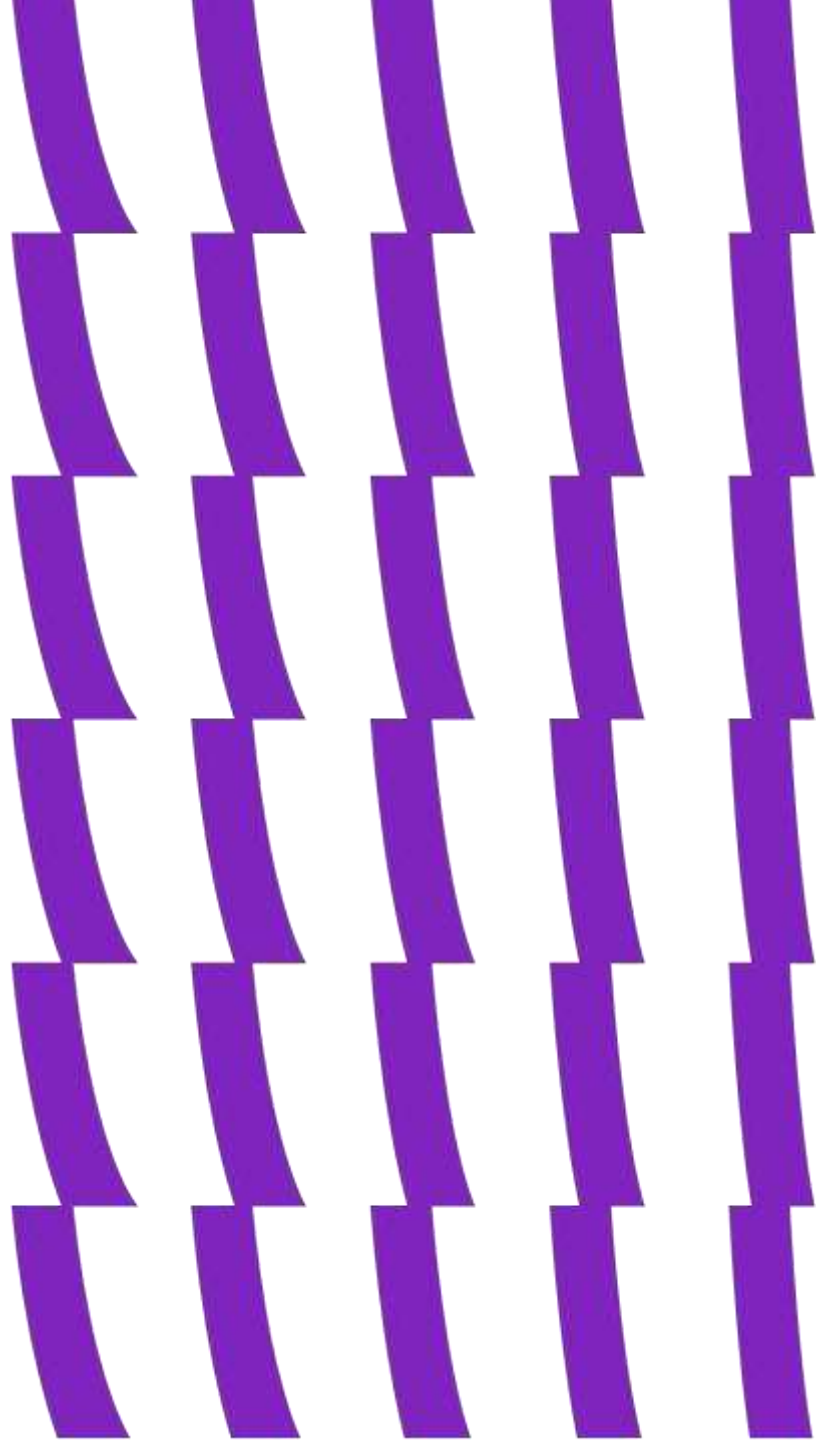
2НФ — Пример нарушения свойств

id_сотр	ФИО	Id_отд	тел	Id_проекта	проект	Id_задания
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

Первичный ключ в отношении: {id_сотр, id_проекта, id_задания}

Аномалии

- Аномалия вставки. В отношение нельзя вставить данные о сотруднике, который пока не участвует ни в одном проекте.
- Аномалия удаления. Если по проекту временно прекращены работы, то при удалении данных о работах по этому проекту будут удалены и данные о самом проекте (наименование проекта). При этом если был сотрудник, который работал только над этим проектом, то будут потеряны и данные об этом сотруднике.
- Аномалия обновления. Если необходимо изменить какую-либо информацию о сотруднике, то придётся изменять значения атрибутов во всех записях.



2НФ — Вот теперь точно она

Id_сотр	ФИО	Id_отд	тел
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

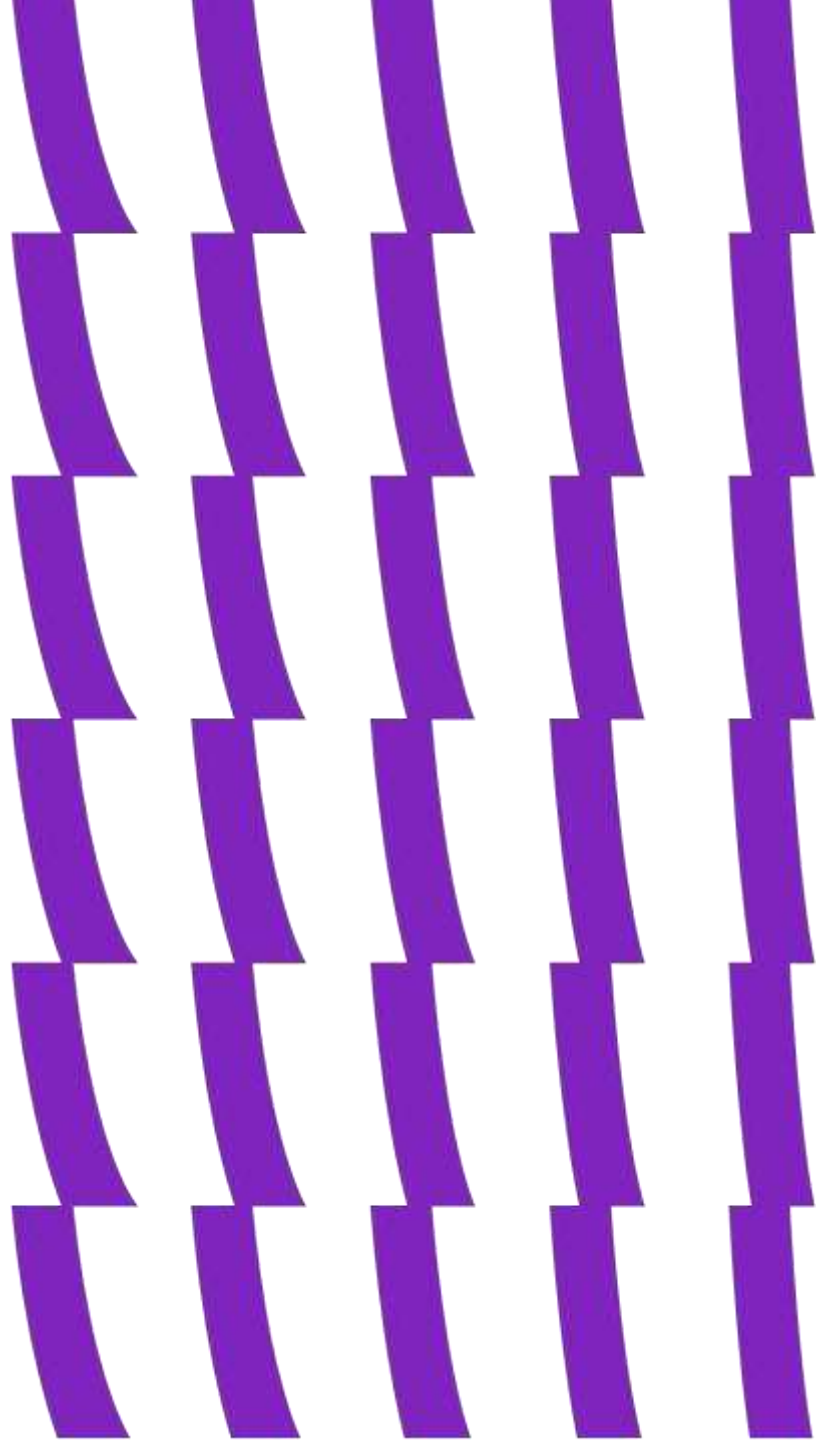
Id_проекта	Проект
1	Космос
2	Климат

Id_сотр	Id_проекта	Id_задания
1	1	1
1	2	1
2	1	2
3	1	3
3	2	2

3НФ

Дадим два эквивалентных определения:

1. 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.
2. Отношение находится в третьей нормальной форме (3НФ) тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимо-независимы.



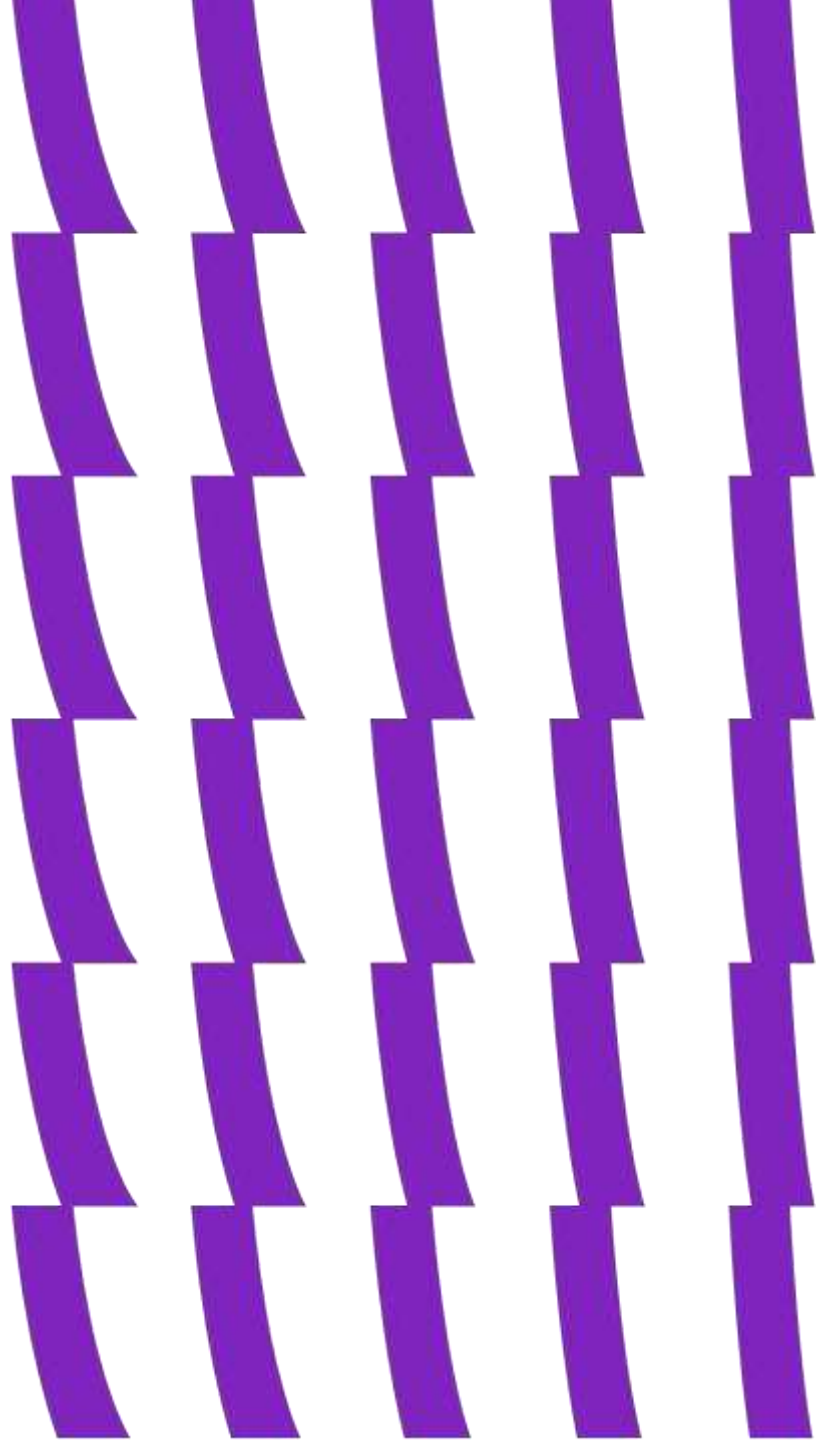
3НФ – Пример нарушения свойств

Id_сотр	ФИО	Id_отд	тел
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Первичный ключ в отношении: {id_сотр}

Аномалии

- Аномалия вставки. Нельзя вставить телефон подразделения, пока в нём не появится хотя бы один сотрудник.
- Аномалия удаления. Если удалить данные о сотруднике, то можно потерять данные об отделе.
- Аномалия обновления. Если необходимо изменить информацию о телефоне подразделения, то необходимо менять много записей.



3НФ

Id_сотр	ФИО	Id_отд	тел
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Id_сотр	ФИО	Id_отд
1	Иванов	1
2	Петров	1
3	Сидоров	2

Id_отд	телефон
1	11-22-33
2	33-22-11

Слабая vs. сильная нормализация

Свойство	Слабая нормализация (1НФ, 2НФ)	Сильная нормализация (3НФ)
Адекватность базы данных предметной области	ХУЖЕ (-)	ЛУЧШЕ (+)
Легкость разработки и сопровождения базы данных	СЛОЖНЕЕ (-)	ЛЕГЧЕ (+)
Скорость выполнения вставки, обновления, удаления	МЕДЛЕННЕЕ (-)	БЫСТРЕЕ (+)
Скорость выборки данных	БЫСТРЕЕ (+)	МЕДЛЕННЕЕ (-)

1НФ, 2НФ, 3НФ — подытожим

Первая нормальная форма (1НФ) — это обычное отношение. Отношение в 1НФ обладает следующими свойствами: в отношении нет одинаковых кортежей; кортежи не упорядочены; атрибуты не упорядочены; все значения атрибутов атомарны и регулярны.

Отношение находится во второй нормальной форме (2НФ) тогда и только тогда, когда отношение находится в 1НФ и нет неключевых атрибутов, зависящих от части сложного ключа.

Отношение находится в третьей нормальной форме (3НФ) тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.

Данные зависят от ключа [1НФ], всего ключа [2НФ] и ничего, кроме ключа [3НФ].

НФБК
(Нормальная форма Бойса — Кодда)

Усиленная третья нормальная форма

Определение 3НФ не совсем подходит для следующих отношений:

1. отношение имеет два или более потенциальных ключа;
2. два и более потенциальных ключа являются составными;
3. они пересекаются, т.е. имеют хотя бы один общий атрибут.

Для отношений, имеющих один потенциальный ключ (первичный), НФБК является 3НФ. Отношение находится в НФБК тогда и только тогда, когда она находится в третьей нормальной форме, и при этом не только любой неключевой атрибут полностью функционально зависит от любого ключа, но и любой ключевой атрибут должен полностью функционально зависеть от любого ключа.

Таким образом, требование о фактической зависимости неключевых атрибутов от всего ключа целиком и ни от чего другого, кроме как от ключа, распространяется и на ключевые атрибуты.

НФБК

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Поставляемое количество VOLUME
1	Фирма 1	1	100
1	Фирма 1	2	200
1	Фирма 1	3	300
2	Фирма 2	1	150
2	Фирма 2	2	250
3	Фирма 3	3	1000

два потенциальных ключа - {PNUM, DNUM} и {PNAME, DNUM}.
PNUM -> PNAME - наименование поставщика зависит от номера поставщика.
PNAME -> PNUM - номер поставщика зависит от наименования поставщика.
{PNUM, DNUM} -> VOLUME - поставляемое количество зависит от первого ключа.
{PNUM,DNUM} -> PNAME - наименование поставщика зависит от первого ключа.
{PNAME, DNUM} -> VOLUME - поставляемое количество зависит от второго ключа.
{PNAME, DNUM} -> PNUM - номер поставщика зависит от второго ключ.

НФБК

Номер поставщика PNUM	Наименование поставщика PNAME
1	Фирма 1
2	Фирма 2
3	Фирма 3

PNUM -> PNAME
PNAME -> PNUM

Номер поставщика PNUM	Номер детали DNUM	Поставляемое количество VOLUME
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	3	1000

{PNUM, DNUM} -> VOLUME

4НФ

Абитуриент	Факультет	Предмет
Иванов	Математический	Математика
Иванов	Математический	Информатика
Иванов	Физический	Математика
Иванов	Физический	Физика
Петров	Математический	Математика
Петров	Математический	Информатика

Единственный ключ
{Абитуриент, Факультет, Предмет}

Аномалия вставки. При попытке добавить в отношение "Абитуриенты-Факультеты-Предметы" новый кортеж, например (Сидоров, Математический, Математика), мы обязаны добавить также и кортеж (Сидоров, Математический, Информатика), т.к. все абитуриенты математического факультета обязаны иметь один и тот же список сдаваемых предметов.

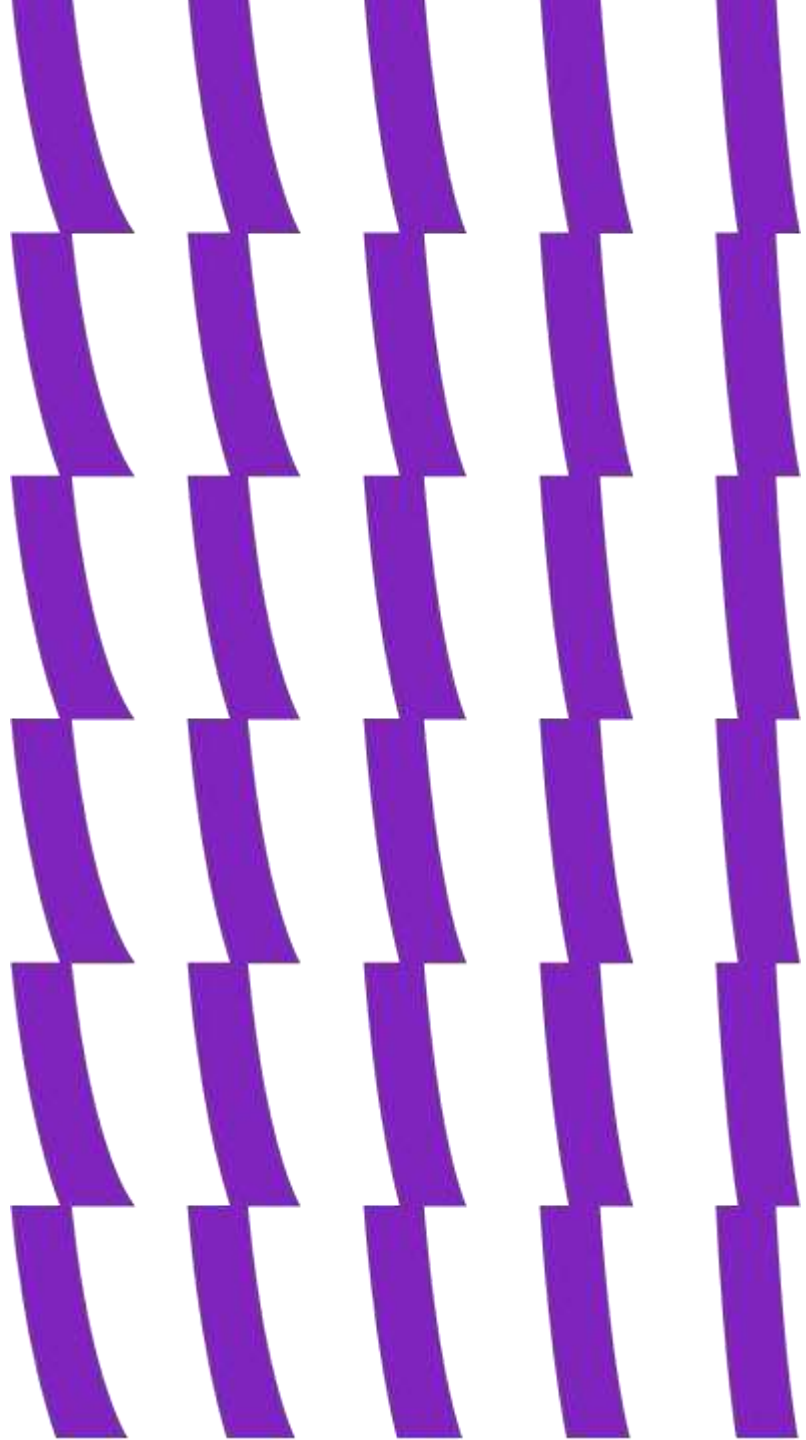
Аномалия удаления. При попытке удалить кортеж (Иванов, Математический, Математика), мы обязаны удалить также и кортеж (Иванов, Математический, Информатика) по той же самой причине

4НФ — Множественная зависимость

Рассмотрим таблицу, в которой есть три непересекающихся набора атрибутов X, Y и Z. Таким образом, требование о фактической зависимости неключевых атрибутов от всего ключа целиком и ни от чего другого, кроме как от ключа, распространяется и на ключевые атрибуты.

Зафиксируем некоторый x_c — и рассмотрим все комбинации x_cyz. Если соответствие между x_c и y никак не зависит от z, то говорят о множественной зависимости X к Y.

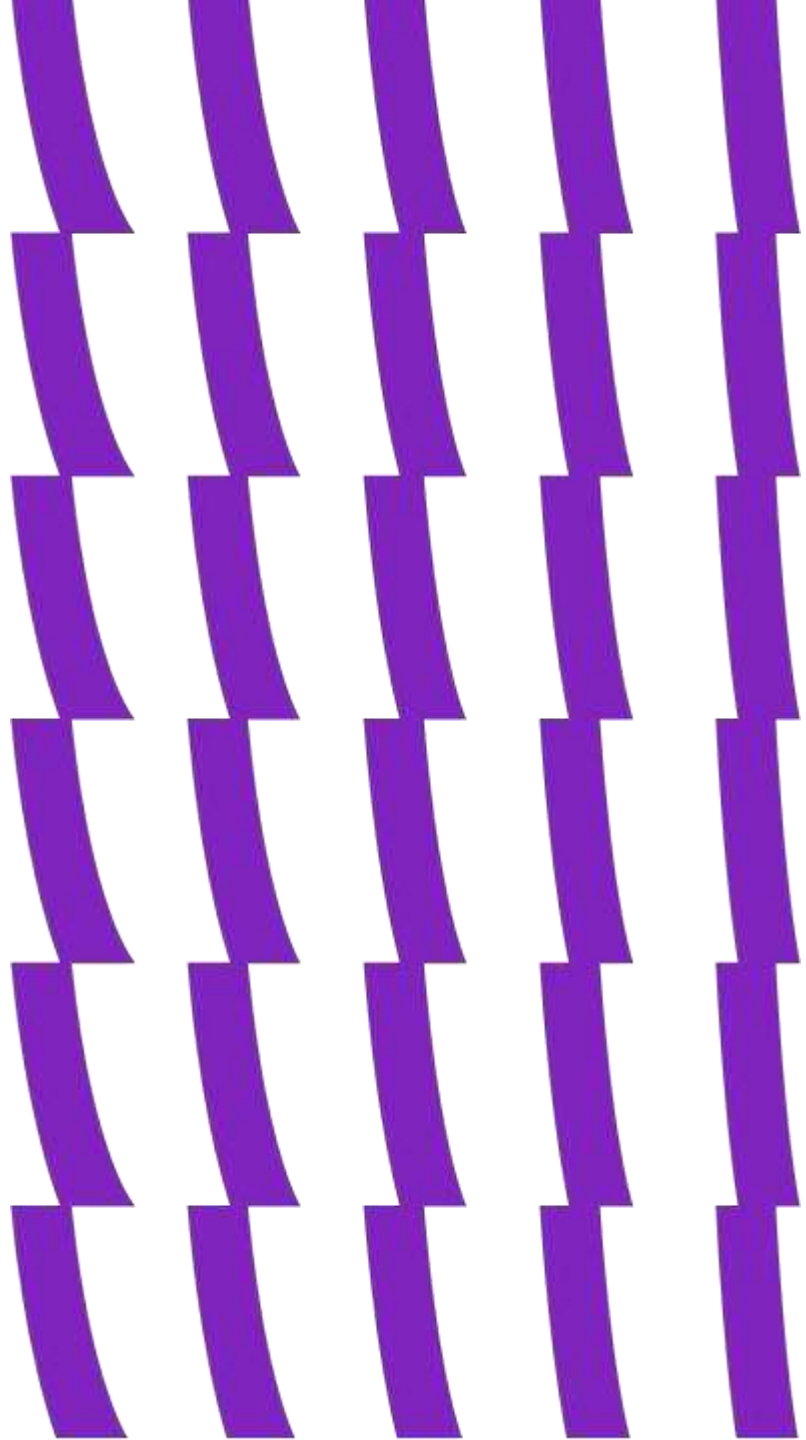
Определение симметрично относительно Y и Z, Поэтому множественные зависимости всегда идут связанными парами и обозначаются как X —>> Y|Z



4НФ

Для каждого факультета (для каждого значения из X) каждый поступающий на него абитуриент (значение из Y) сдает один и тот же список предметов (набор значений из Z), и для каждого факультета (для каждого значения из X) каждый сдаваемый на факультете экзамен (значение из Z) сдается одним и тем же списком абитуриентов (набор значений из Y).

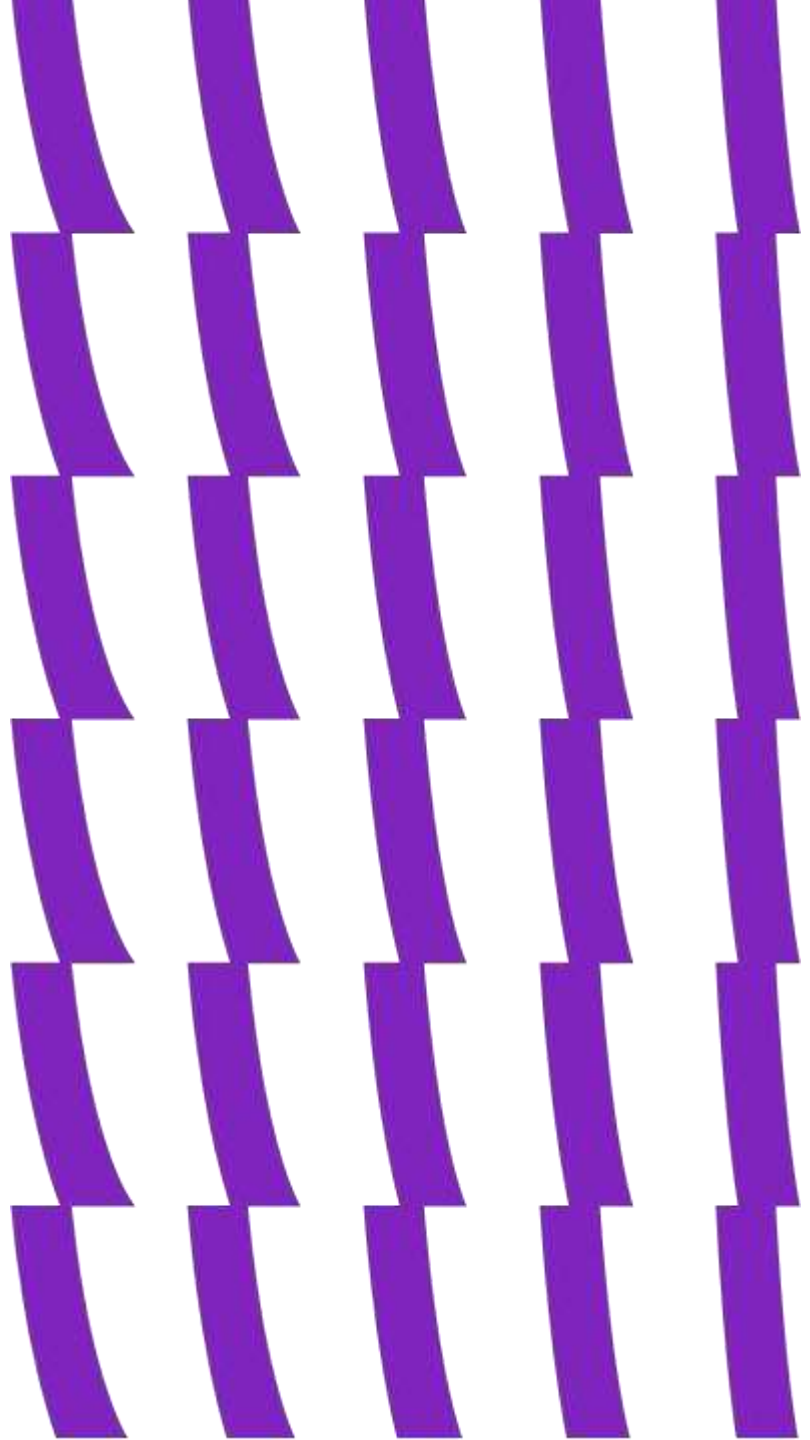
Именно наличие этой зависимости не позволяет независимо вставлять и удалять кортежи. Кортежи обязаны вставляться и удаляться одновременно целыми наборами.



4НФ — Долгожданное определение

Многозначная зависимость X —>> Y|Z называется тривиальной, если одна «сторона» является подмножеством другой.

Отношение находится в четвертой нормальной форме (4НФ) тогда и только тогда, когда отношение находится в НФБК и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от её потенциальных ключей.

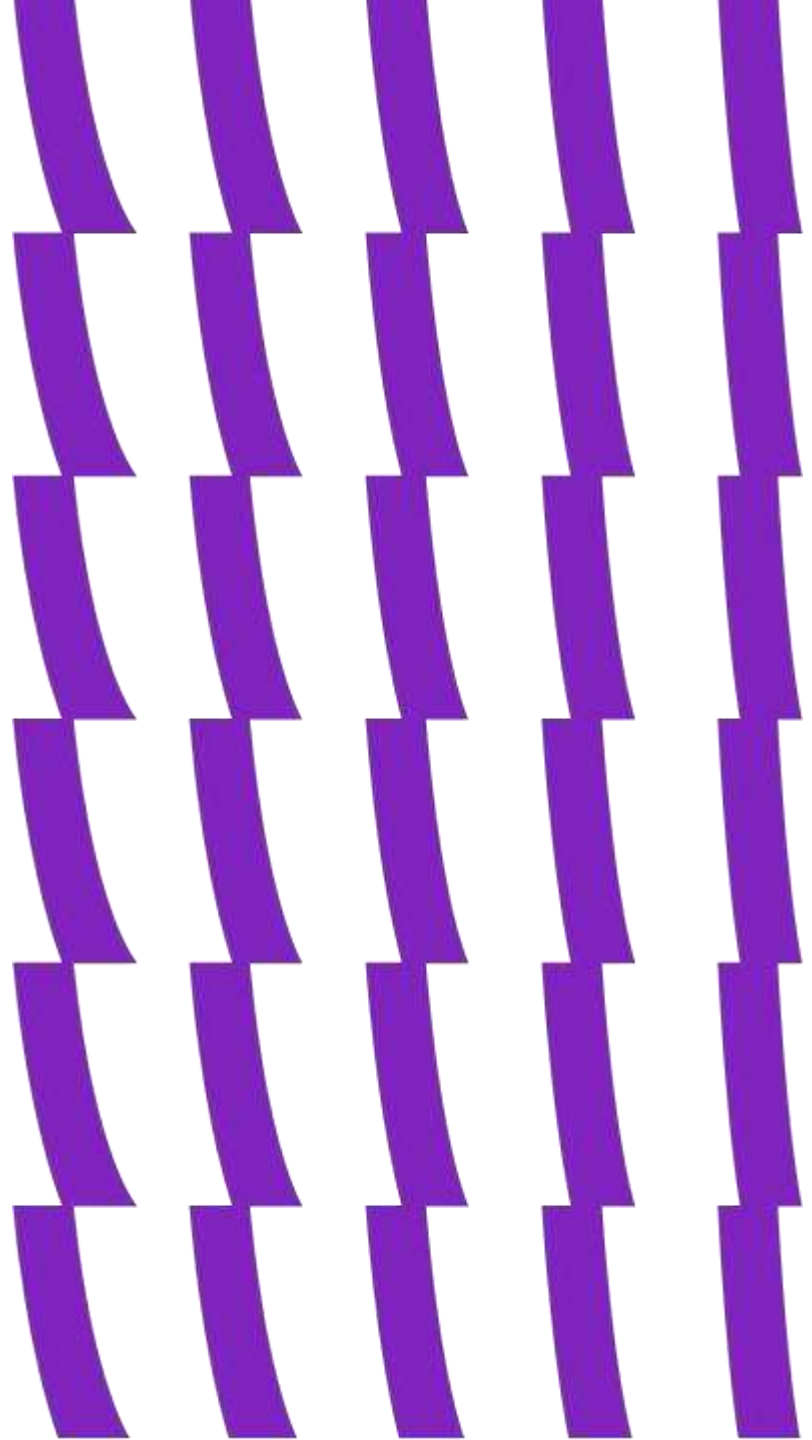


4НФ

Абитуриент	Факультет
Иванов	Математический
Иванов	Физический
Петров	Математический

Факультет	Предмет
Математический	Математика
Математический	Информатика
Физический	Математика
Физический	Физика

С помощью Natural join можем восстановить исходную сущность



5НФ

Ключ: {Предмет, Лектор, Семестр}
Таблица в 4НФ

Формальное описание имеющихся данных в таблице может не соответствовать скрытым бизнес-требованиям о природе этих данных.

Предмет	Лектор	Семестр
Физика	Иванов	1
Физика	Петров	1
Математика	Петров	1
Математика	Петров	2
БД	Дубин	2
БД	Ковриков	2
БД	Дубин	3

Дополнительные требования к таблице:

- Иванов – только физика
- Физика – только на 1 курсе
- Физика – только Иванов и Петров

5НФ

Отношениям в 4НФ, при наличии некоторых ограничений, свойственны аномалии обновления. Эти аномалии невозможно устранить путем проецирования отношения на две проекции, требуется декомпозиция на три или большее число отношений.

R = R[A] NATURAL JOIN R[B] NATURAL JOIN …. R[Z]

Зависимость соединения называется тривиальной зависимостью соединения, если одно из подмножеств атрибутов совпадает со всем множеством атрибутов отношения.

Отношение находится в пятой нормальной форме (5НФ) тогда и только тогда, когда оно находится в 4НФ и отсутствуют сложные зависимые соединения между атрибутами

5НФ

Ключ: {Предмет, Лектор, Семестр}
Таблица в 4НФ

Нетривиальная зависимость соединения:(
{Лектор, Семестр},
{Предмет, Лектор},
{Лектор, Семестр}
)

Однако ни одно из подмножеств не содержит ключа целиком.

Предмет	Лектор	Семестр
Физика	Иванов	1
Физика	Петров	1
Математика	Петров	1
Математика	Петров	2
БД	Дубин	2
БД	Ковриков	2
БД	Дубин	3
БД	Ковриков	3

Избыточность: При добавление Коврикова, добавляй и БД

5НФ

Предмет - Лектор

Предмет	Лектор
Физика	Иванов
Физика	Петров
Математика	Петров
БД	Дубин
БД	Ковриков

Лектор - Семестр

Лектор	Семестр
Иванов	1
Петров	1
Петров	2
Дубин	2
Дубин	3
Ковриков	2

Предмет - Семестр

Предмет	Семестр
Физика	1
Математика	1
Математика	2
БД	1
БД	2
БД	3

НФБК, 4НФ, 5НФ

Отношение находится в НФБК, когда каждая нетривиальная и неприводимая слева функциональная зависимость обладает потенциальным ключом в качестве детерминанта.

Обобщением 3НФ на случай, когда отношение имеет более одного потенциального ключа, является нормальная форма Бойса — Кодда.

Отношение находится в четвертой нормальной форме (4НФ) тогда и только тогда, когда отношение находится в НФБК и не содержит нетривиальных многозначных зависимостей.

Отношение находится в пятой нормальной форме (5НФ) тогда и только тогда, когда любая имеющаяся зависимость соединения является тривиальной.

6НФ. Строка содержит первичный ключ и не более одного другого атрибута

6НФ

Отношение находится в шестой нормальной форме тогда и только тогда, когда оно удовлетворяет всем нетривиальным зависимостям соединения.

Т.е. отношение находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6НФ, также находится и в 5НФ.

ID	Начало	Конец	Тема встречи
1	10:20	11:00	Обсуждение статуса задач
2	15:40	18:20	Разработка архитектуры проекта
3	19:20	19:55	Подведение итогов Q1

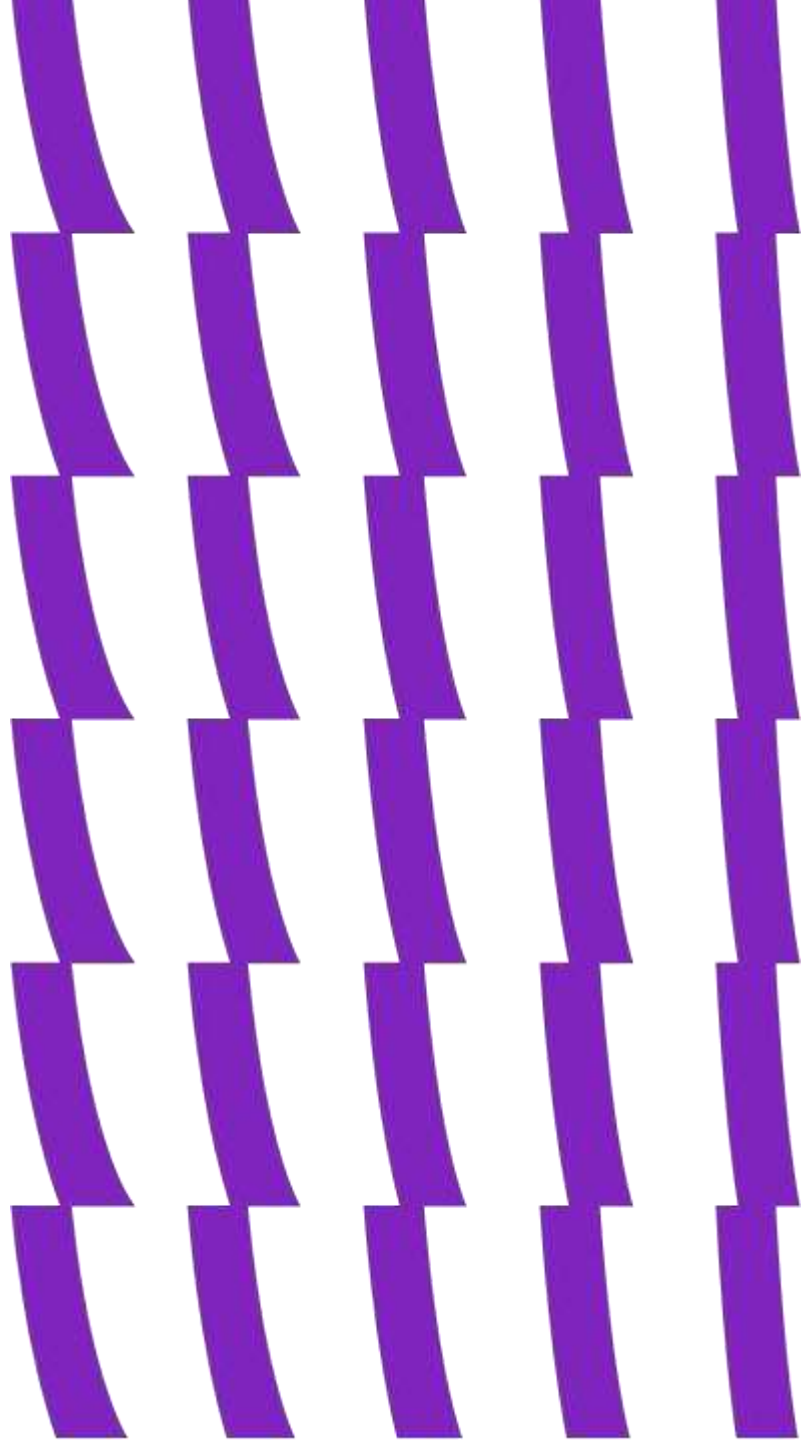
ID	Начало
1	10:20
2	15:40
3	19:20

ID	Конец
1	11:00
2	18:20
3	19:55

ID	Тема встречи
1	Обсуждение статуса задач
2	Разработка архитектуры проекта
3	Подведение итогов Q1

Резюме по блоку

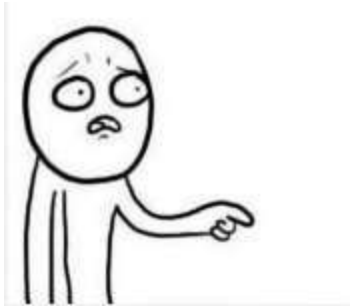
- Нормализация – процесс декомпозиции отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы
- 1НФ, 2НФ — денормализованные отношения, используются в хранилищах данных для ускорения доступа
- 3НФ, НФБК — классические модели по схеме снежинка\звезда
- НФБК, 4НФ — подход Data Vault
- 5НФ, 6НФ — подход Anchor modelling



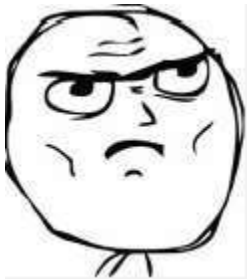
Хранилище данных

Как это бывает

- Мы большая крутая контора,
нам пора строить свое хранилище с красивыми отчетами и Data Science!



- Но нужно же сначала проанализировать…



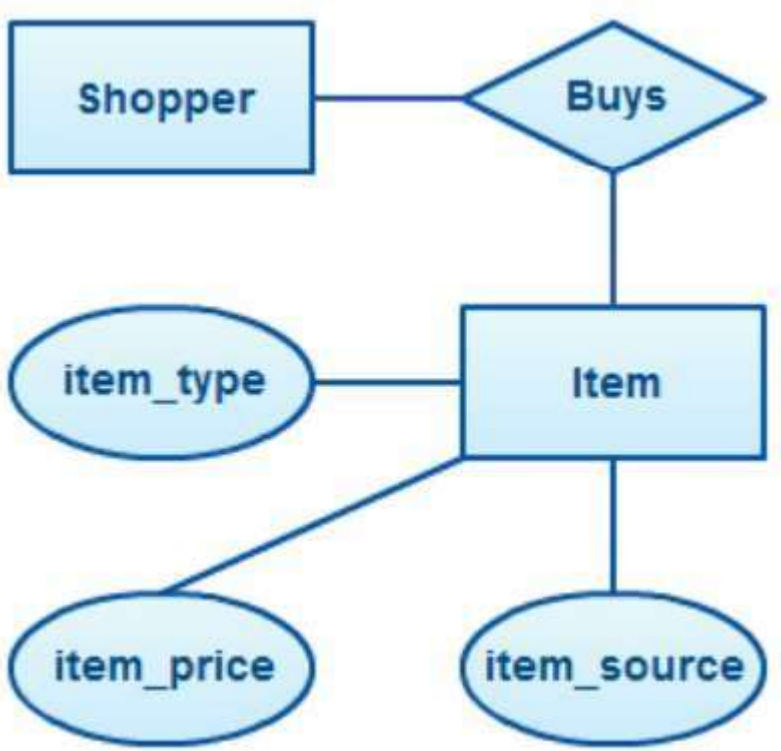
- Некогда думать, надо делать!

С чего начать?

С проектирования модели хранилища.

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

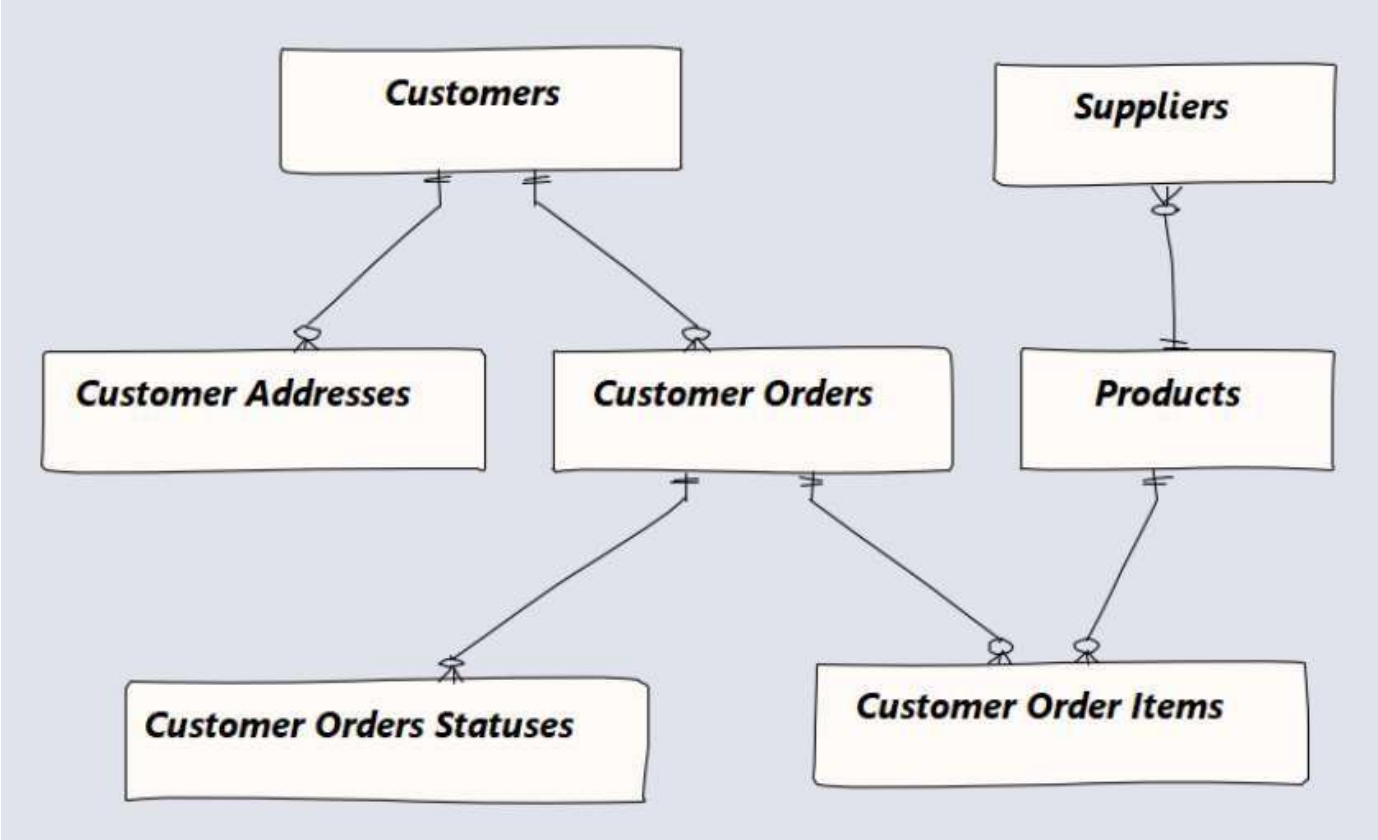
1. Концептуальной модели
2. Логической модели
3. Физической модели



Концептуальная модель данных

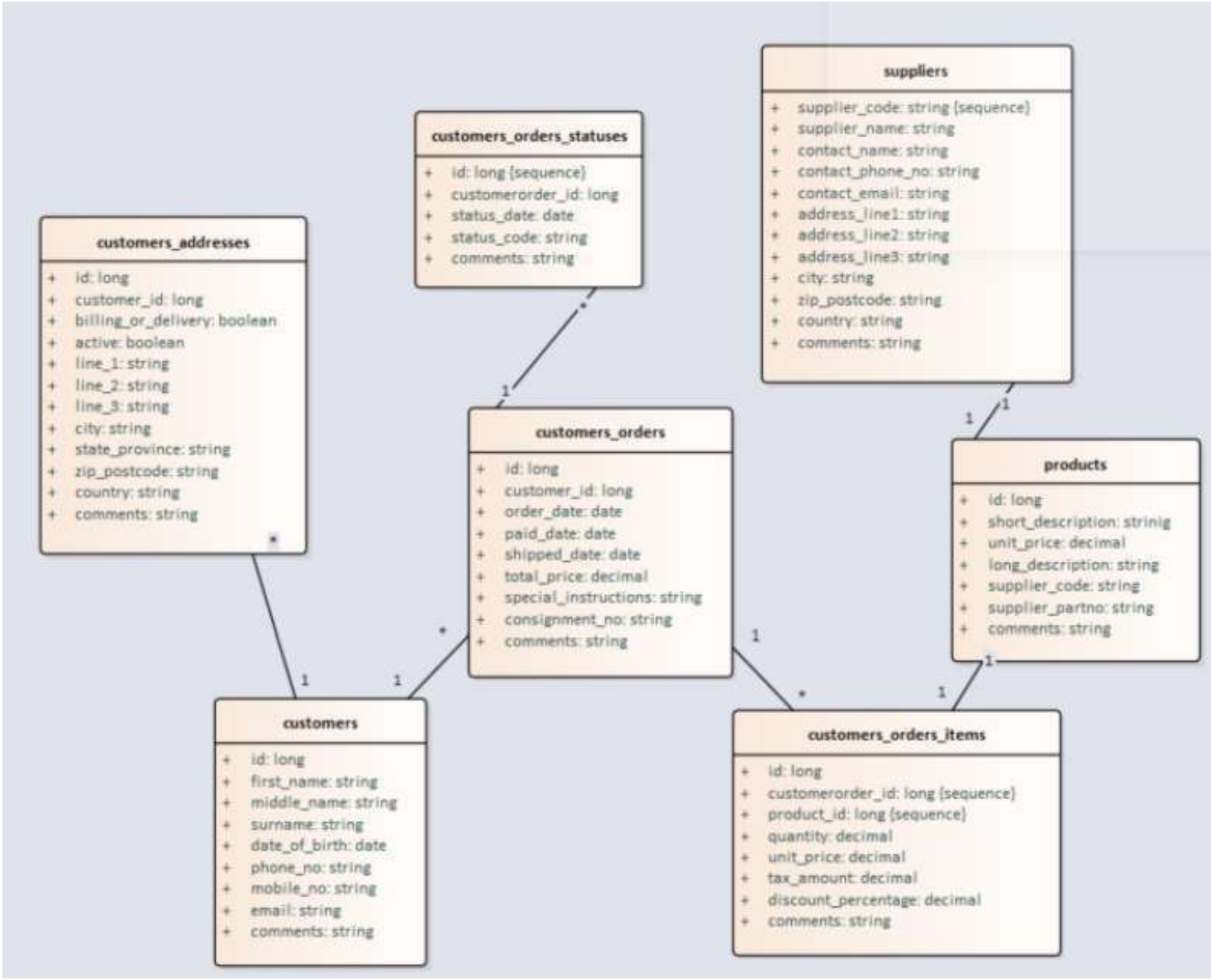
Концептуальная модель представляет собой описание основных сущностей и отношений между ними

Она является отражением предметных областей, в рамках которых планируется построение DWH



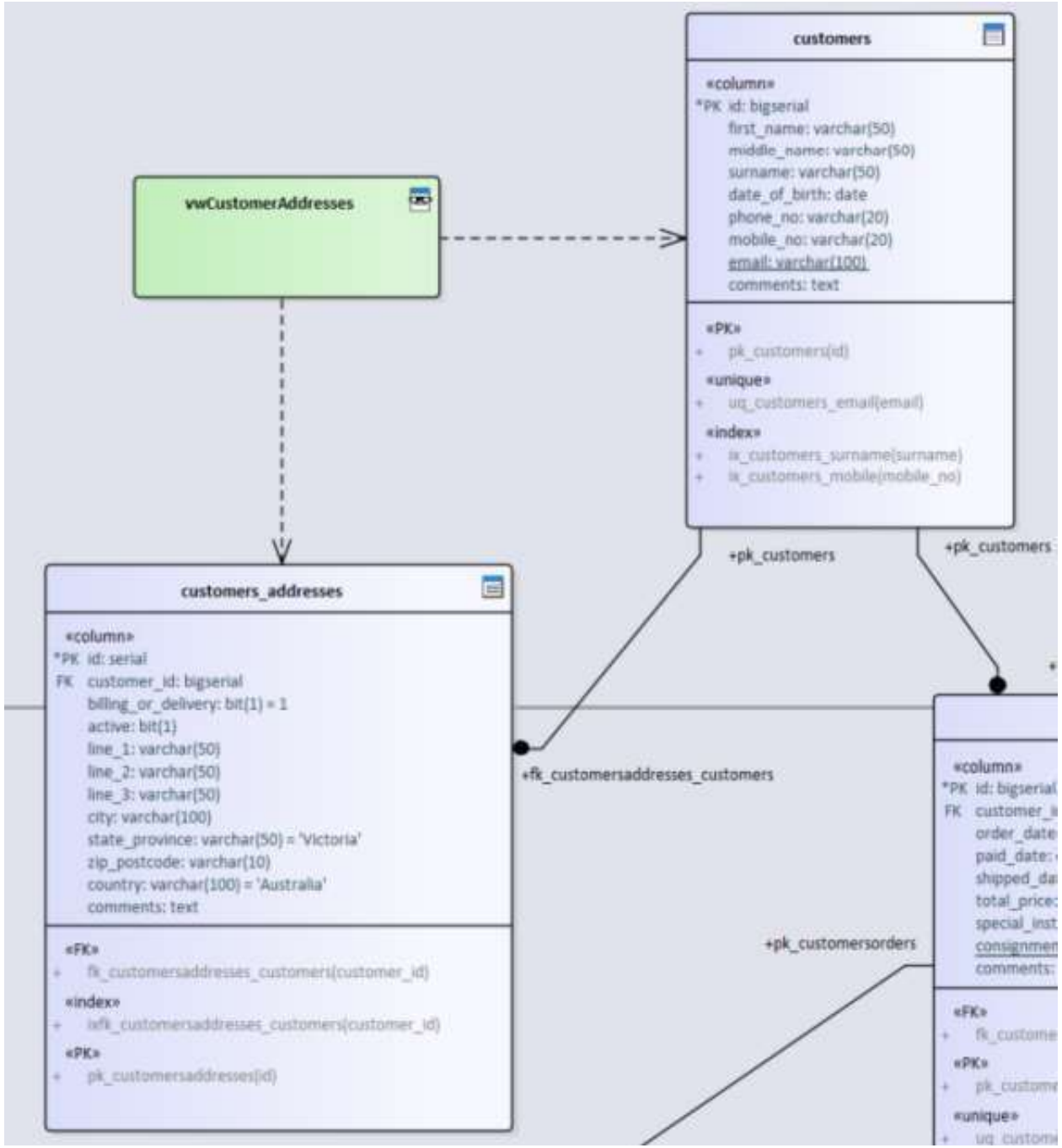
Логическая модель данных

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними



Физическая модель данных

Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных



KIMBALL *VS* INMON

Билл Инмон

Билл Инмон - отец системного подхода в области анализа и проектирования DWH.

Автор классического определения хранилищ данных. Другое название его подхода к проектированию - Corporate Information Factory (CIF)

Подход Инмона

Проектирование логической модели “сверху вниз”.

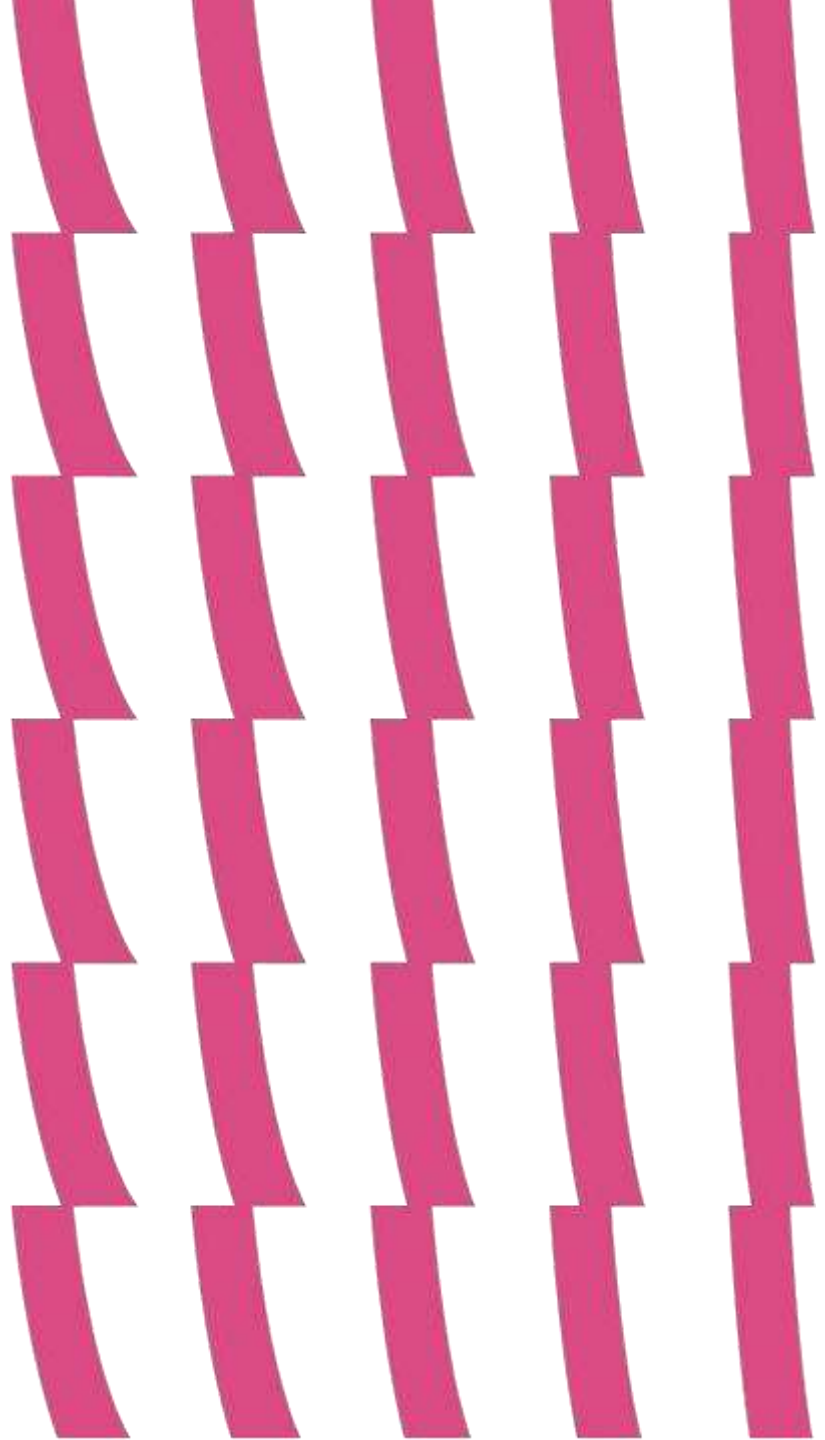
1. Тщательный анализ бизнеса в целом

2. Выявляются бизнес-области

3. В них - ключевые бизнес-сущности

4. Затем - их характеристики (атрибуты) и связи между ними.

В результате анализа появляется понимание, какие сущности участвуют в бизнес-процессах и как они взаимодействуют друг с другом.

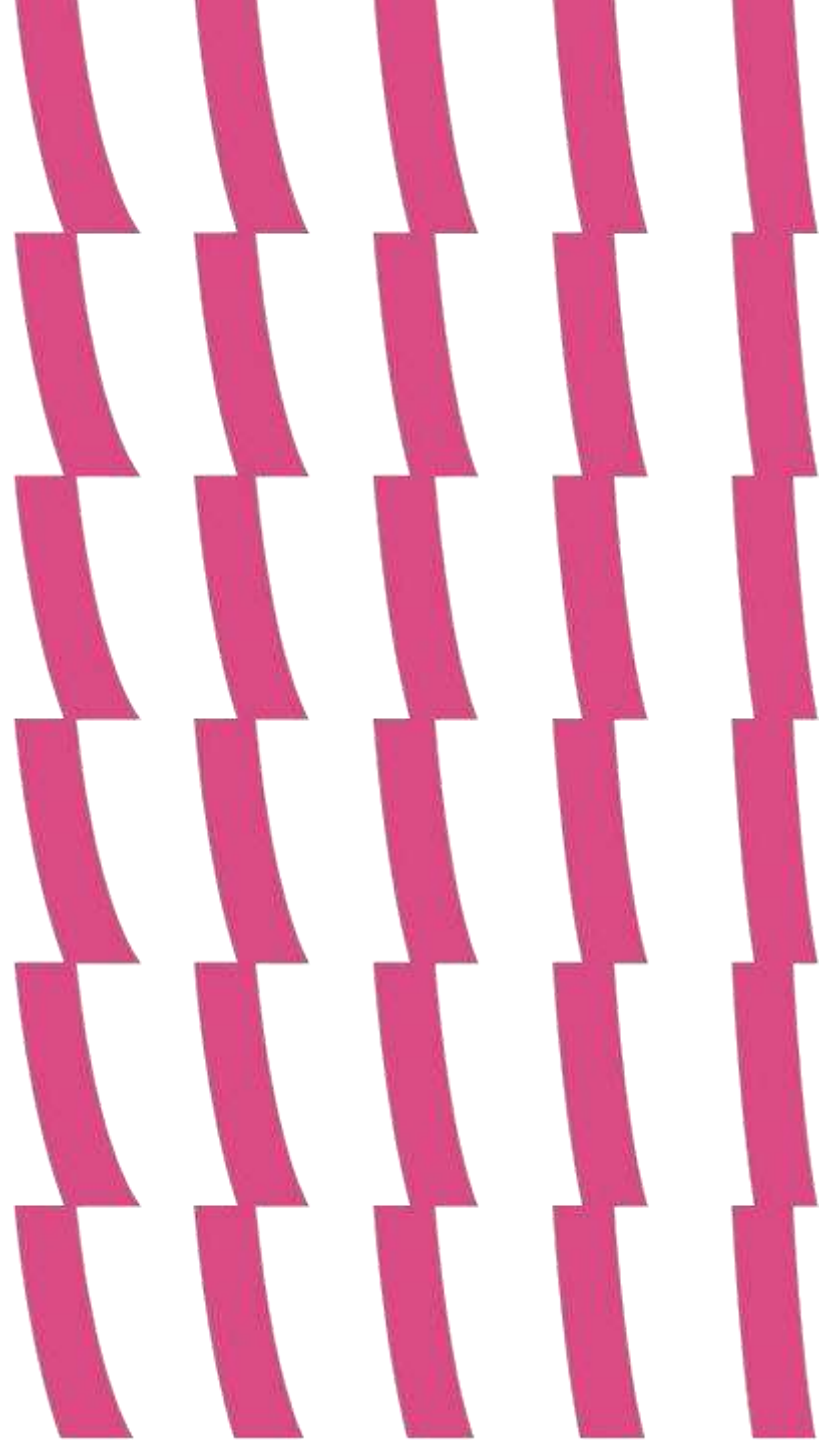


Подход Инмона

Следующий шаг – это разработка физической модели на основе логической.

- Можно не завершать логическую модель полностью, а переносить ее в базу по частям - отдельными сущностями.
- Модель должна быть в нормализованном виде.

Полученная в результате разработки нормализованная структура по Инмону называется хранилищем данных или детальным слоем



Физическая модель. Одна версия правды

STG -> DDS -> EMART

Нисходящий подход

Система источник — внешний источник, откуда данные собираются независимо от типа данных. Данные могут быть структурированными, полуструктурированными и неструктурированными.

Поскольку данные, извлеченные из внешних источников, не соответствуют определенному формату, необходимо проверить эти данные для загрузки в хранилище данных. Для этого рекомендуется использовать инструмент ETL .

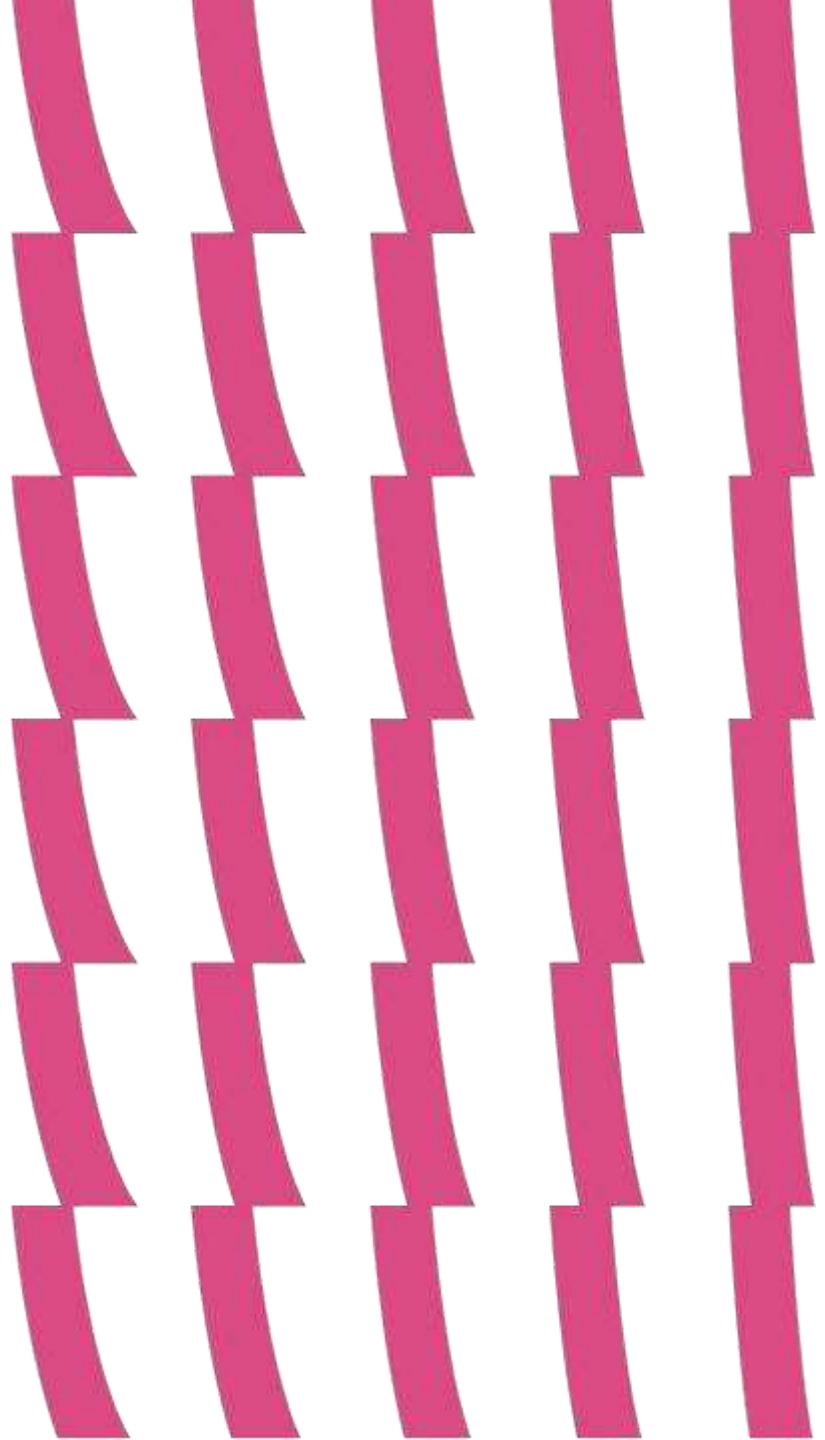
- E (extract): данные извлекаются из внешнего источника данных.
- T (Transform): данные преобразуются в стандартный формат.
- L (Load): данные загружаются в хранилище данных после преобразования их в стандартный формат.

После очистки данных они сохраняются в хранилище данных в качестве центрального хранилища.

Витрины данных — Data Mart также является частью компонента хранения. Хранит информацию об определенной функции организации, которая обрабатывается одним органом. В организации может быть как можно больше витрин данных в зависимости от функций. Можно также сказать, что витрина данных содержит подмножество данных, хранящихся в хранилище данных.

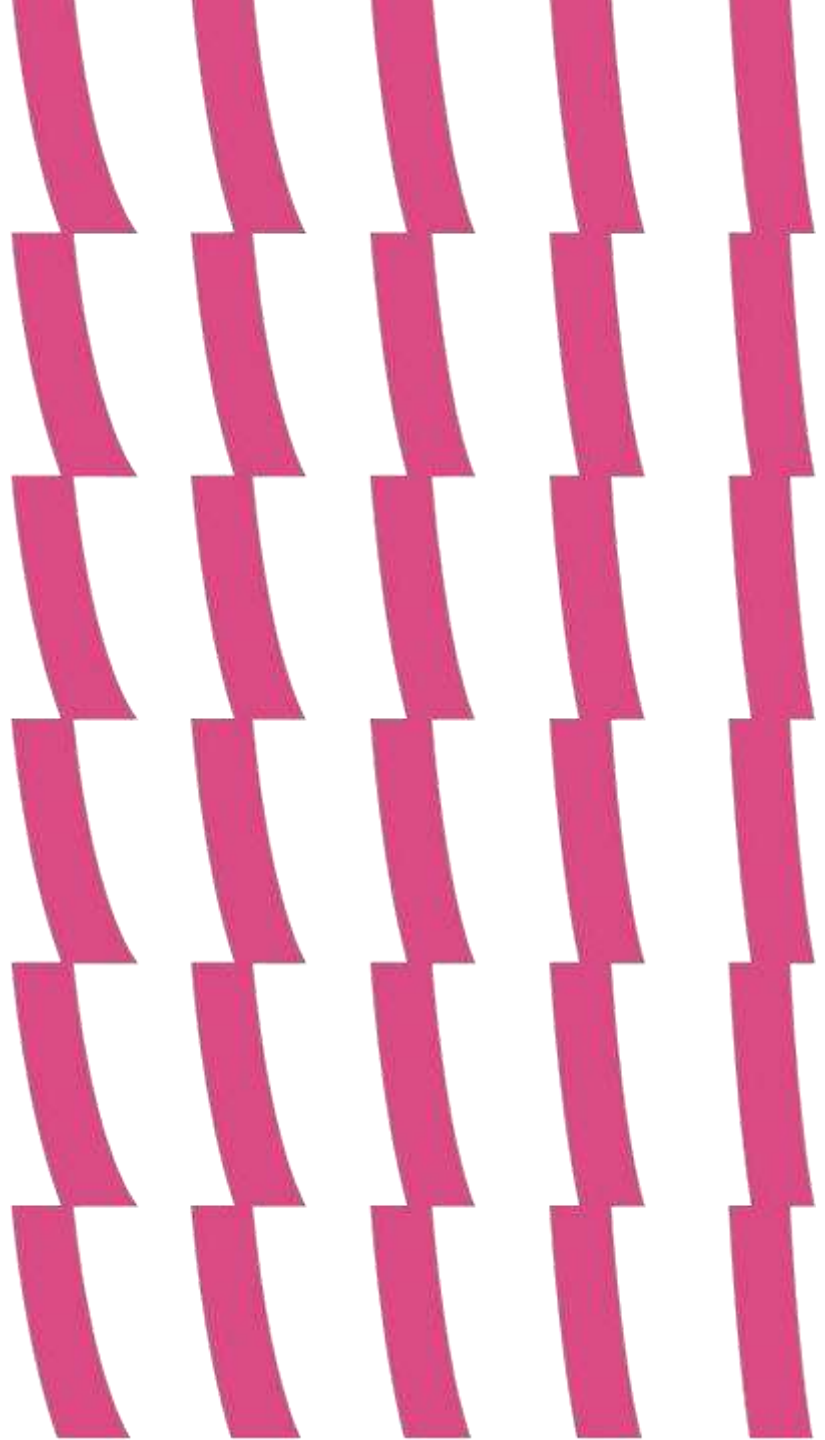
Характеристики подхода Инмона

- Использование реляционной модели организации атомарных данных и пространственной - для организации суммарных данных.
- Разработка хранилища не сразу, а по частям. Позволяет при необходимости вносить изменения в небольшие блоки данных
- Использование 3NF (и выше) для организации детальных данных (предоставляет широкие возможности для манипулирования ими, изменения формата и способа представления).
- DWH-I - это проект корпоративного масштаба, охватывающий все отделы и обслуживающий нужды всех пользователей корпорации.
- DWH-I - это не механическая коллекция витрин данных, а физически целостный объект.



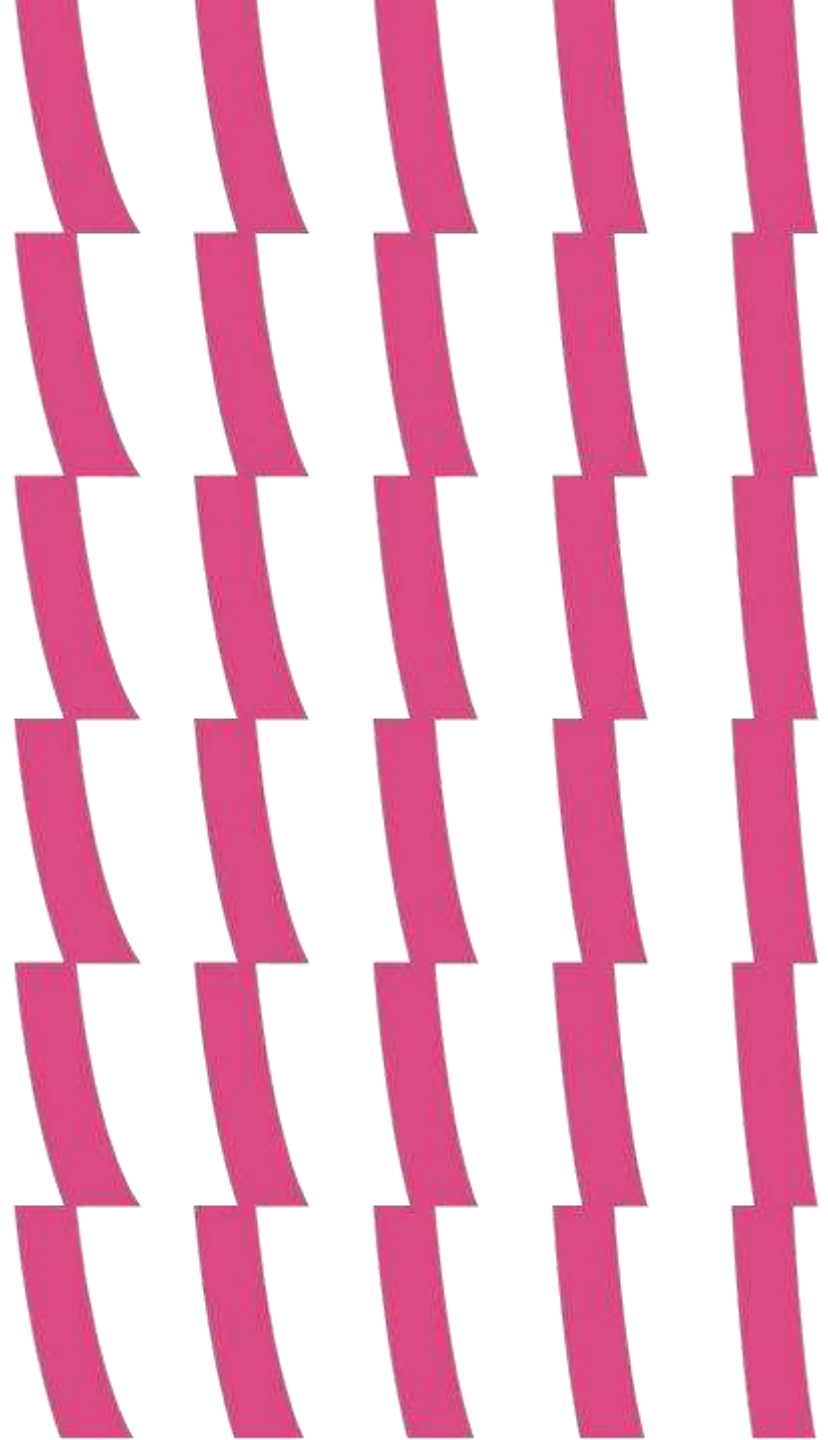
Плюсы подхода Инмона

- Единая версия правды
- Отсутствие избыточности в данных упрощает и стандартизирует ETL процессы
- Отсутствие избыточности в данных снижает вероятность появления аномалий (противоречивости) в данных при загрузке
- Логическая модель представляет собой проекцию бизнеса и процессов в нем, поэтому запросы заказчика довольно легко интерпретировать и встроить в общую концепцию
- Гибкость – за счет высокой нормализации легко добавлять новые сущности и связи между ними
- Поддерживаются задачи самого высокого спектра, так как детальный слой содержит наиболее полную информацию



Минусы подхода Инмона

- Много джойнов
- Долго (то есть - дольше, чем по Кимбаллу)
- Больше ETL’я – выше вероятность ошибки
- Для поддержки и разработки такой модели необходимы более высококвалифицированные кадры, которых иногда тяжело найти
- Латентность увеличивается (данные нужно сначала разложить по полочкам, потом поджойнить для получения чего-либо понятного)

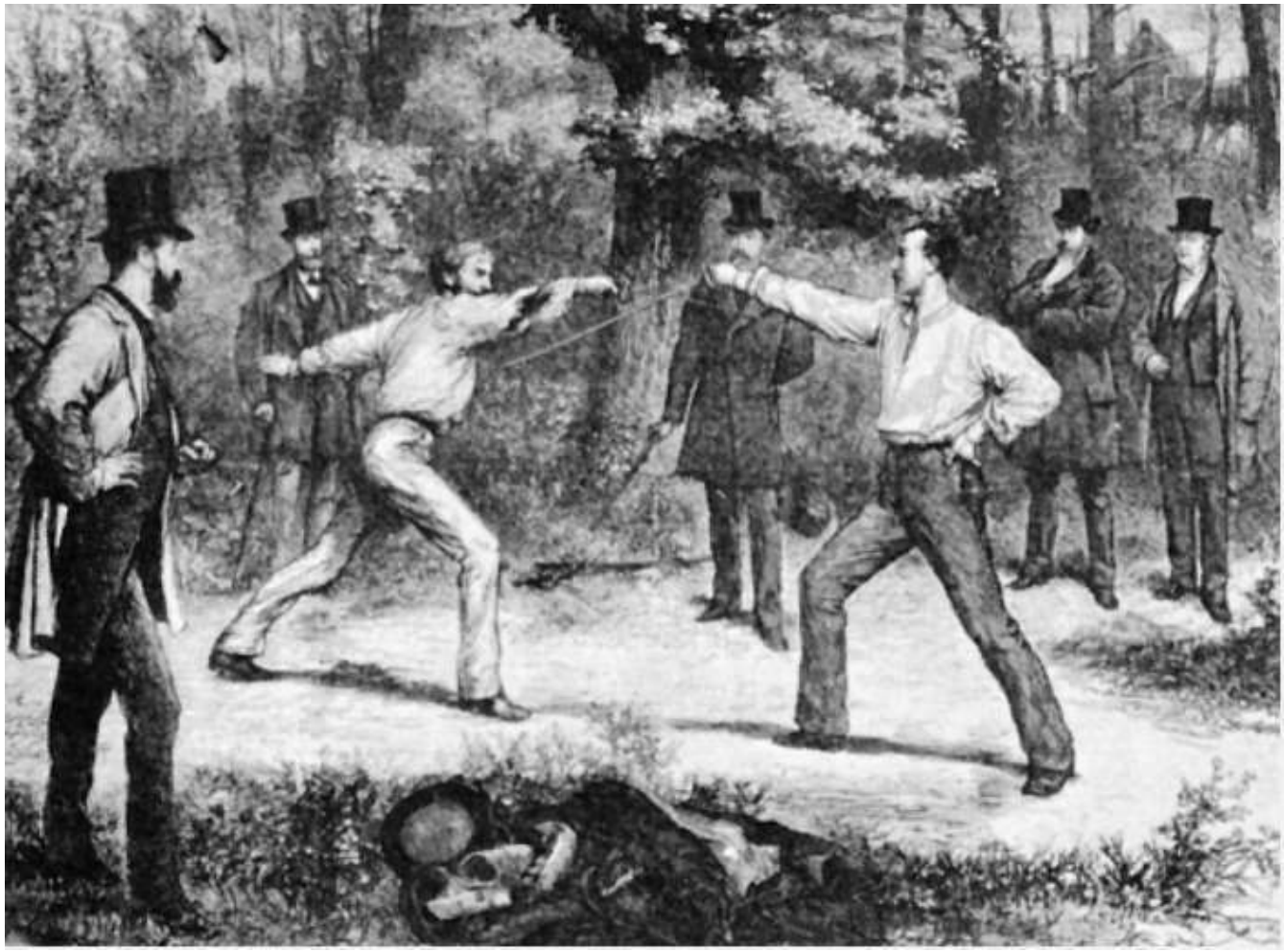


Ральф Кимбалл и Dimensional Architecture

Ральф Кимбалл – автор стандарта де факто в области DWH.

Другие названия его подхода к проектированию - Kimball dimensional architecture, пространственная модель, хранилище с архитектурой шины.

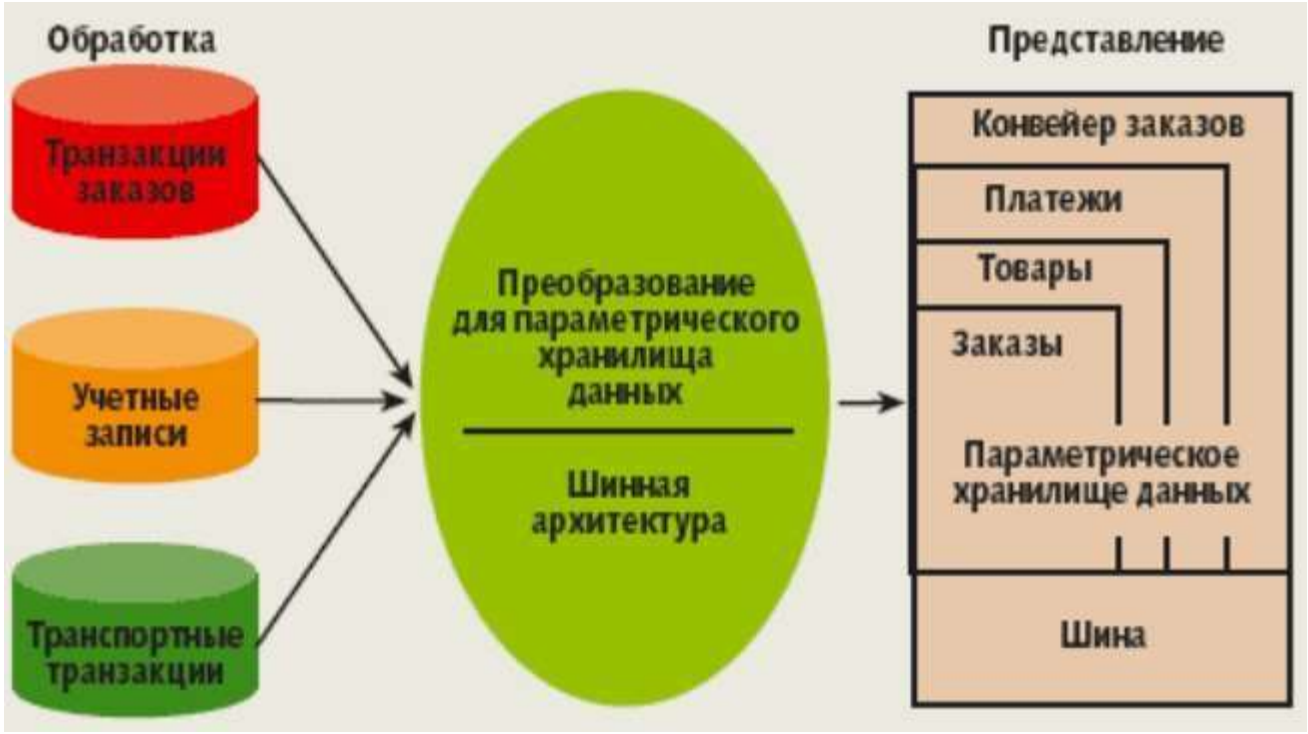
Подход «снизу вверх»



Хранилище по Кимбаллу

DWH по Кимбаллу - копия транзакционных данных, специально структурированных для запроса и анализа.

Таким образом, хранилище по Кимбаллу можно назвать коллекцией витрин данных (отчетов).

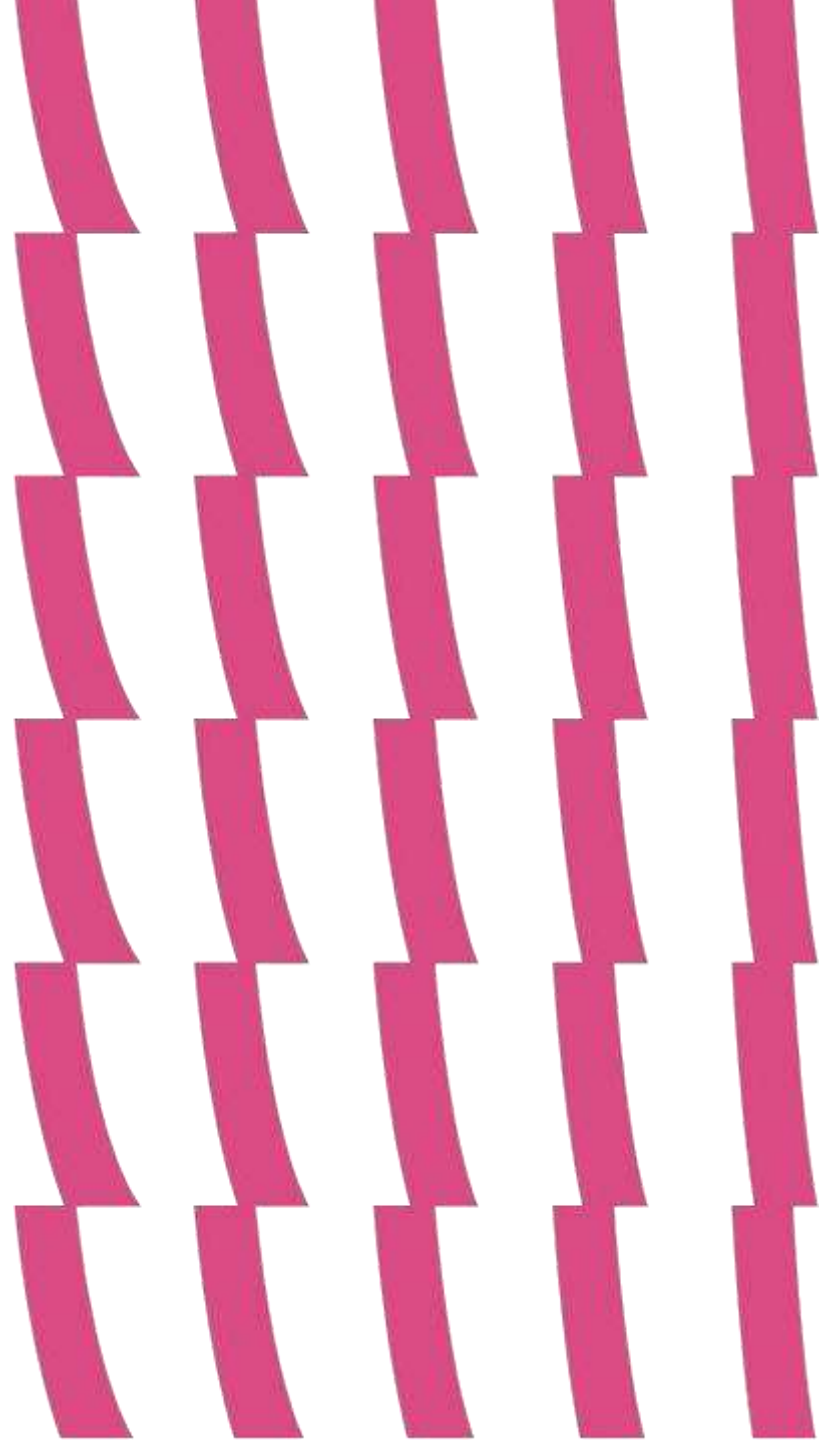


Хранилище по Кимбаллу

- Анализ – узнаем, какие отчеты нужны
- Смотрим, в каких источниках есть эти данные
- Создаем витрины
- Первичные данные из источников преобразуются в информацию, пригодную для использования, на этапе подготовки данных.

Обязательно принимаются во внимание требования к скорости обработки информации и качеству данных

Фокус на процессах



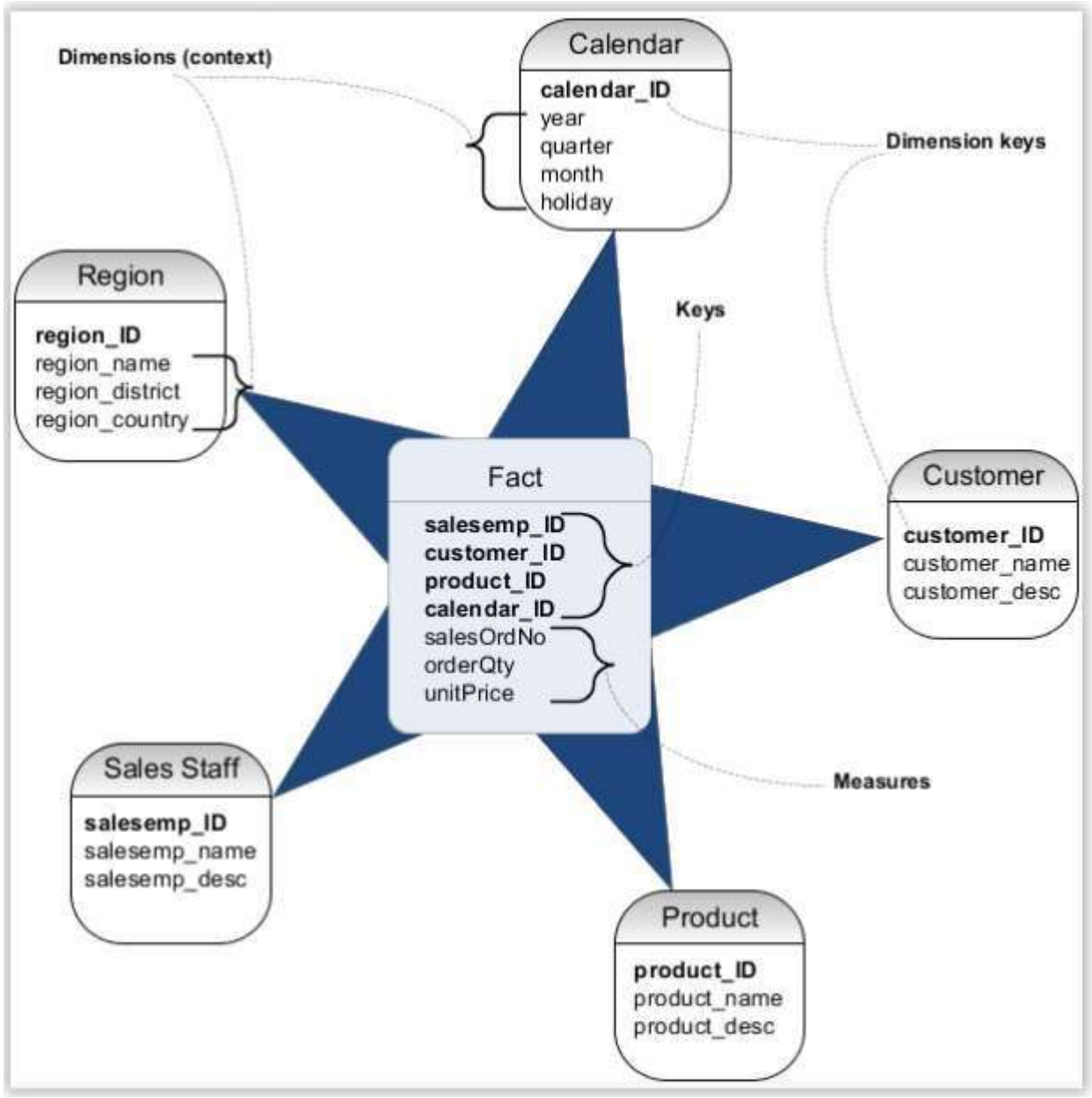
Модель типа “звезда”

Из стейджингового слоя данные попадают в модель типа «звезда»

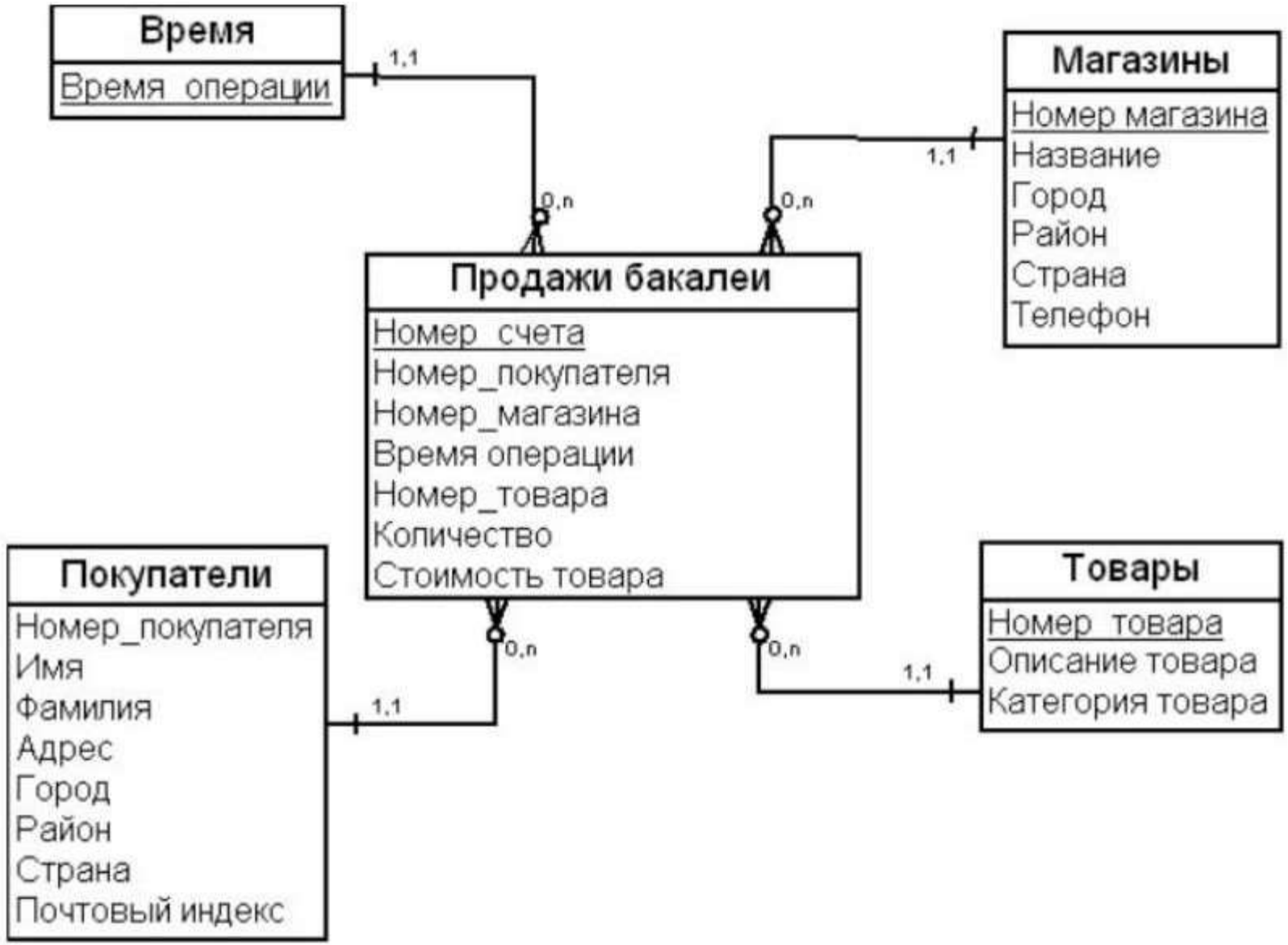
Эта модель представляется таблицами фактов и измерений.

Таблица фактов содержит метрики, числа

Таблица измерений содержит описания для метрик, чисел



Модель типа “звезда”. Пример



Матрица хранилища (Enterprise Bus Matrix)

Матрица корпоративного хранилища - это таблица или документ, которая описывает связи между таблицами фактов и измерений.

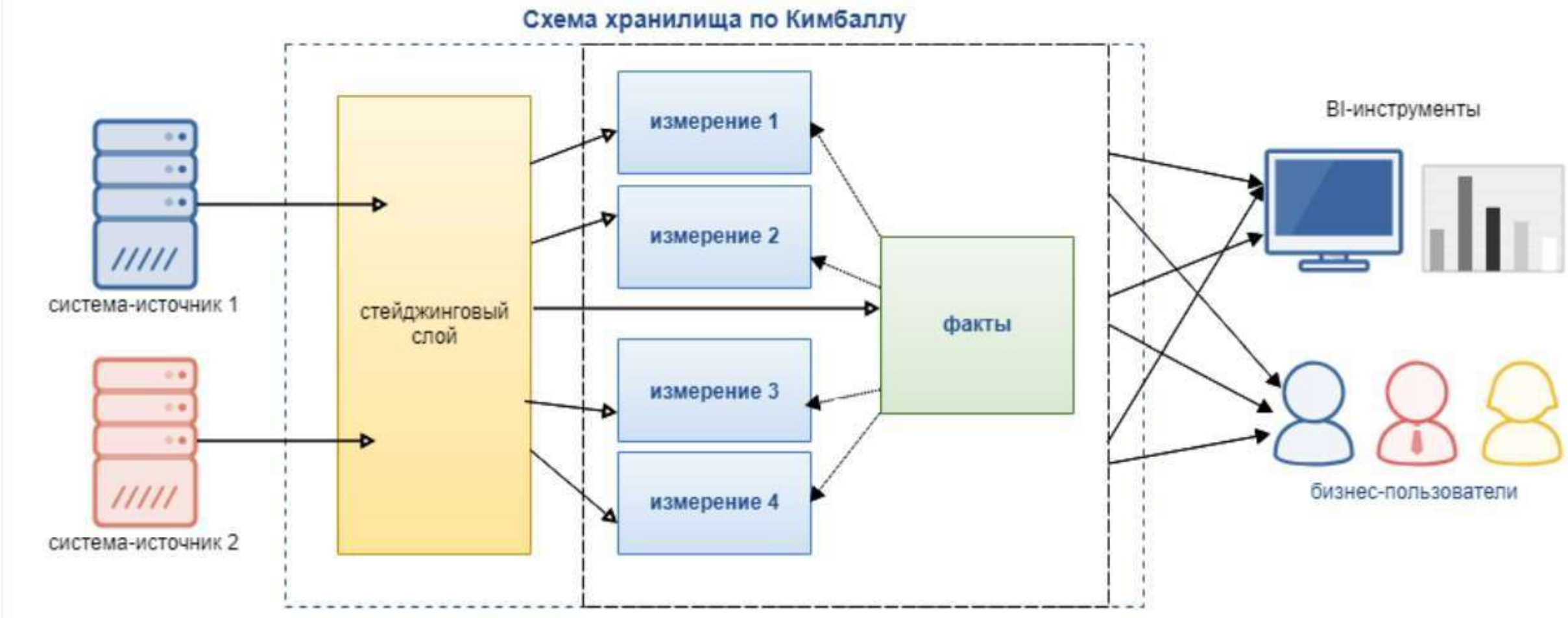
Матрица может быть отнесена к категории гибридных моделей, являющихся инструментом технического проектирования, инструментом управления частями и инструментом обмена данными

Бизнес-процессы	Измерения							
	Дата	Продукт	Производитель	Пункт распределения	Грузоотправитель	Склад	Клиент	Промо-акция
Заявки на закупки	X	X	X	X				
Пункт распределения – доставка	X	X	X	X	X			
Пункт распределения – хранение	X	X		X				
Склад – доставка	X	X		X	X	X		
Склад – хранение	X	X				X		
Склад – продажа	X	X				X	X	X
Возвраты	X	X				X	X	X

По сравнению с Инмоном

STG ->ВИТРИНЫ

Подход Кимбалла - суть



DWH по Кимбаллу - набор витрин данных.
Детального слоя в понимании Инмона в этой модели нет, поэтому первые результаты можно получить достаточно быстро.

Типичные черты подхода Ральфа Кимбалла

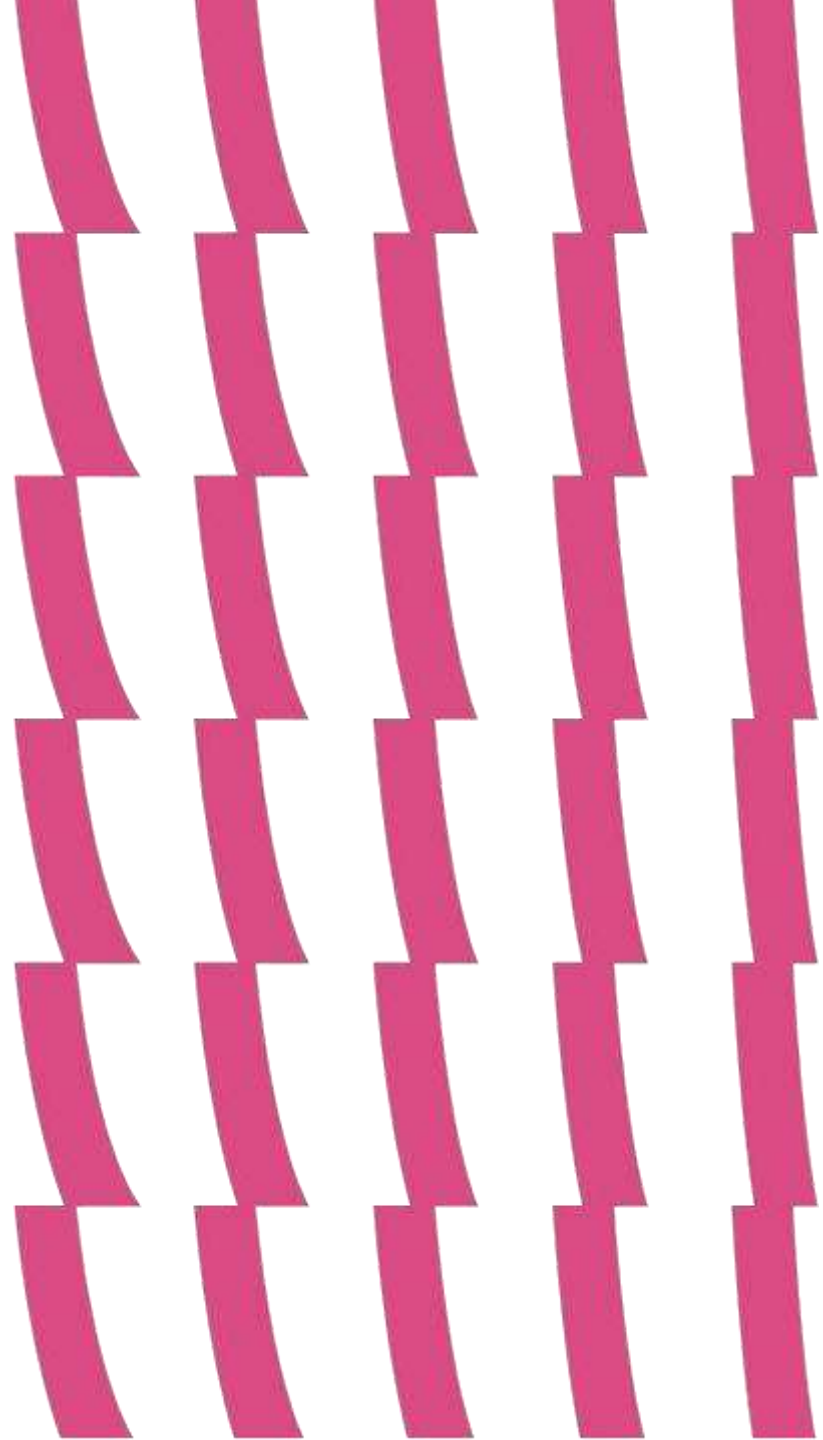
1. Использование модели данных с архитектурой "звезда"

2. Использование двухуровневой архитектуры (стадия подготовки данных + Хранилище данных с архитектурой шины).

3. DWH-K обладает следующими характеристиками:

- фокус на процессах;
- включает как данные о транзакциях, так и агрегаты;
- включает витрины данных, посвященные только одной предметной области;

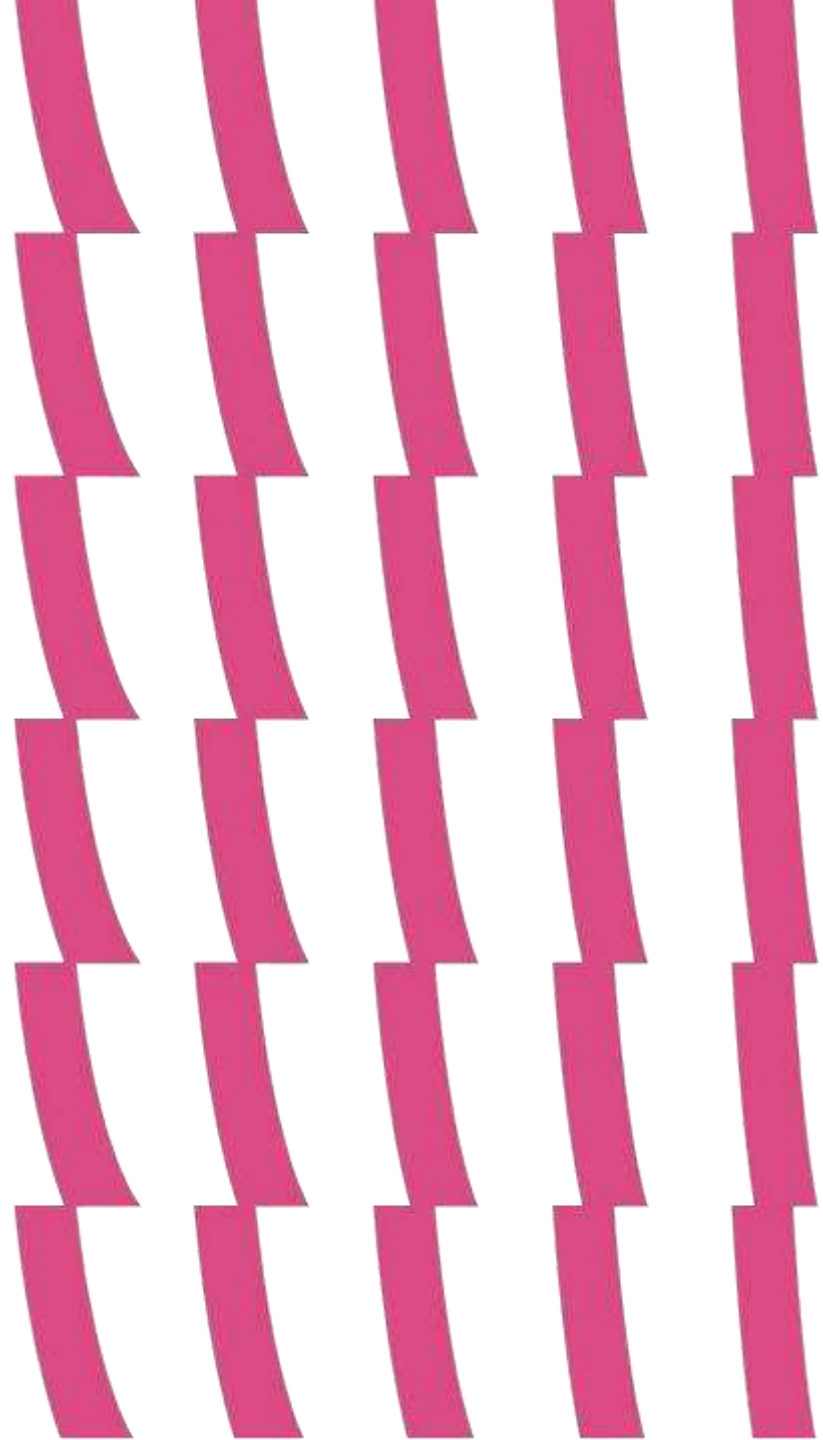
4. DWH-K не является единым физическим репозиторием (в отличие от DWH-I). Это коллекция витрин данных, каждая из которых имеет архитектуру типа "звезда".



Фундаментальные отличия подходов Инмона и Кимбалла

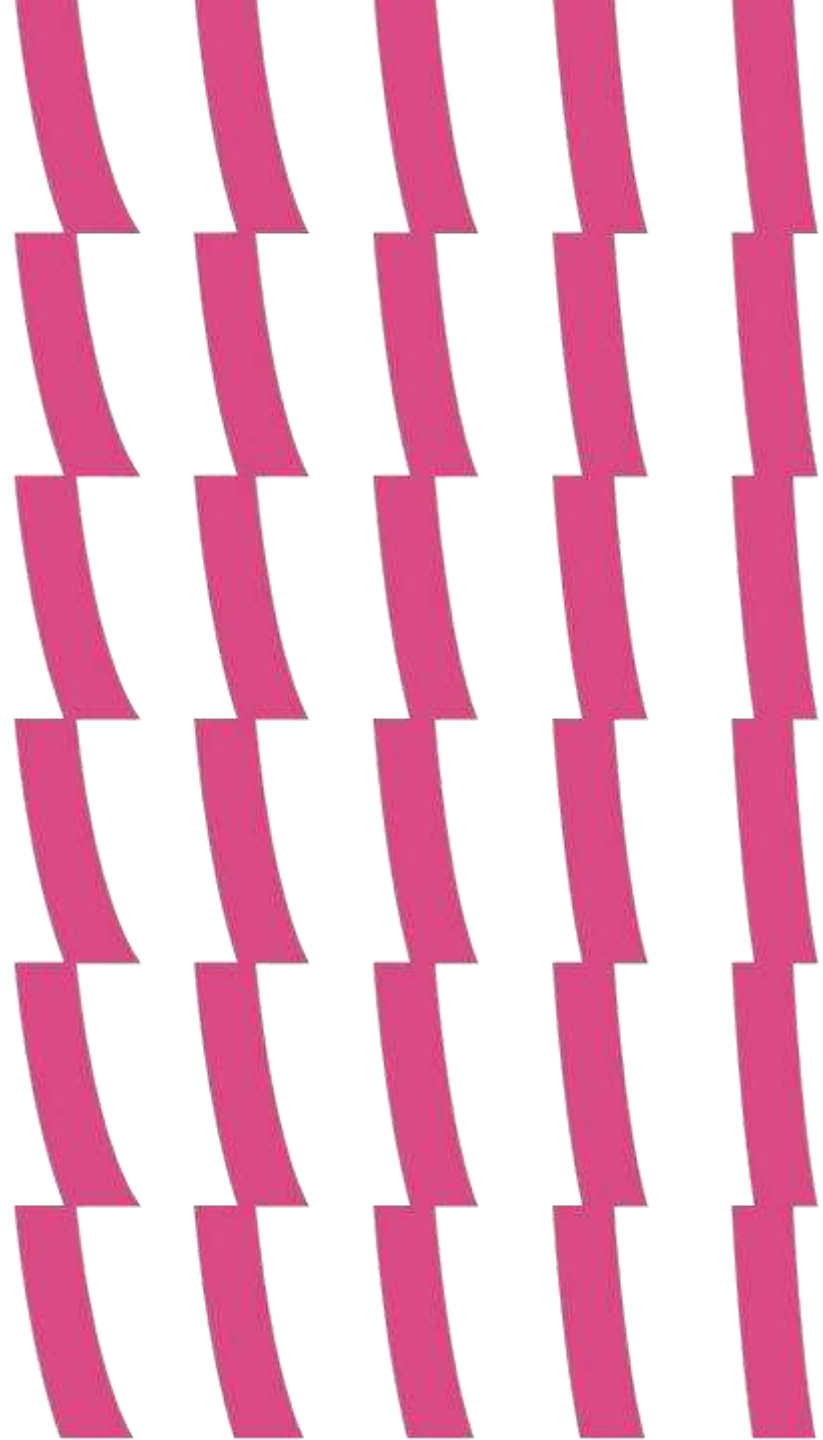
В DWH-K, в отличие от DWH-I:

- Централизованного хранилища как такового нет
- Пространственные модели поддерживают бизнес-процессы, а не бизнес-сущности
- При построении хранилища используются денормализованные структуры



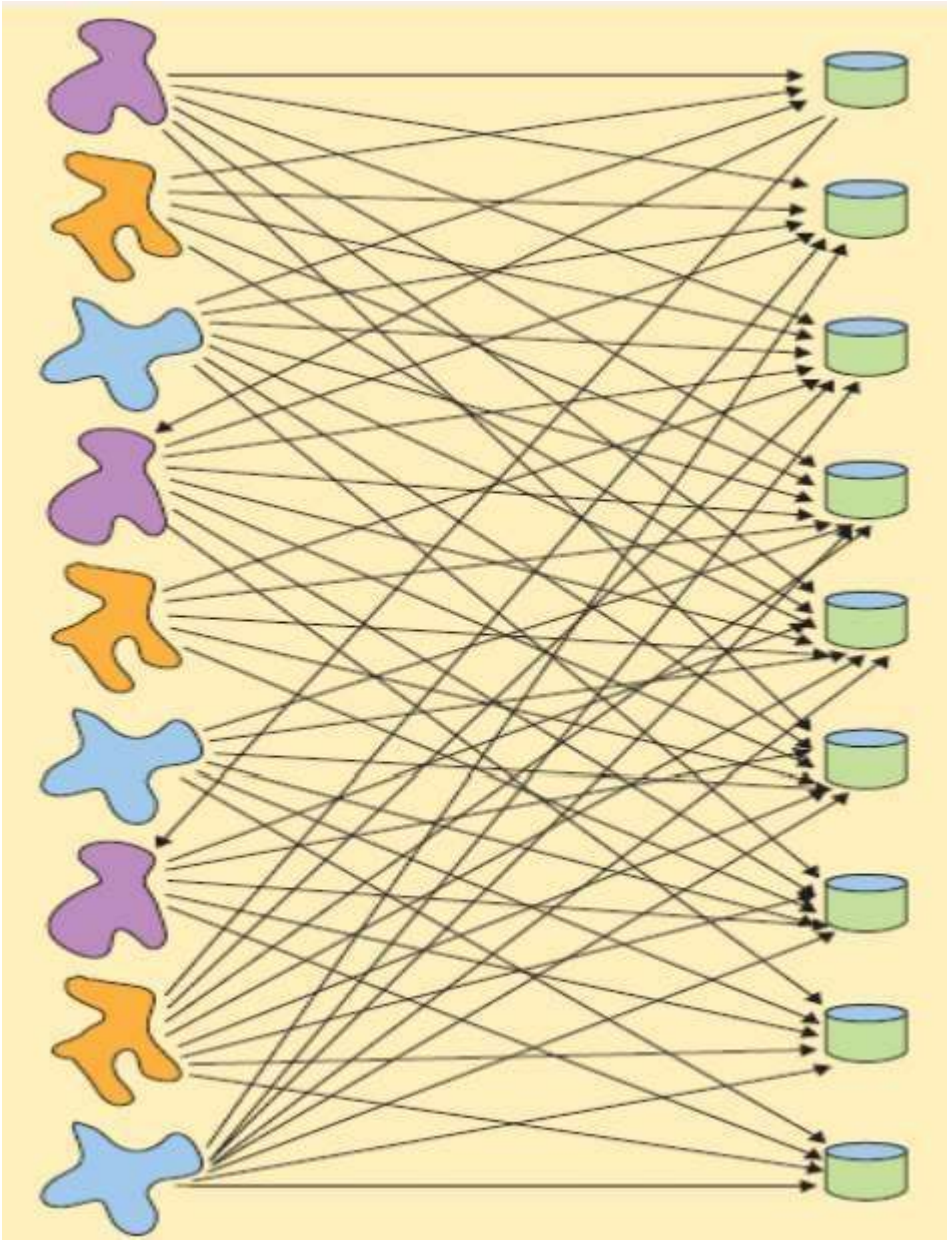
Плюсы подхода Кимбалла

1. Хранилище может быть построено достаточно быстро с «нуля» (т.к. не надо заниматься нормализацией)
2. При небольшом количестве сущностей логическая и физическая модель хранилища гораздо проще, содержит меньше связей.
3. Простота и понятность. Бизнес-пользователям достаточно хорошего знания SQL и матрицы хранилища перед глазами
4. Для получения одной сущности и ее взаимосвязей не требуется много джойнов
5. Для поддержки и доработки хранилища не требуется большой высококвалифицированной команды



Минусы подхода по Кимбаллу

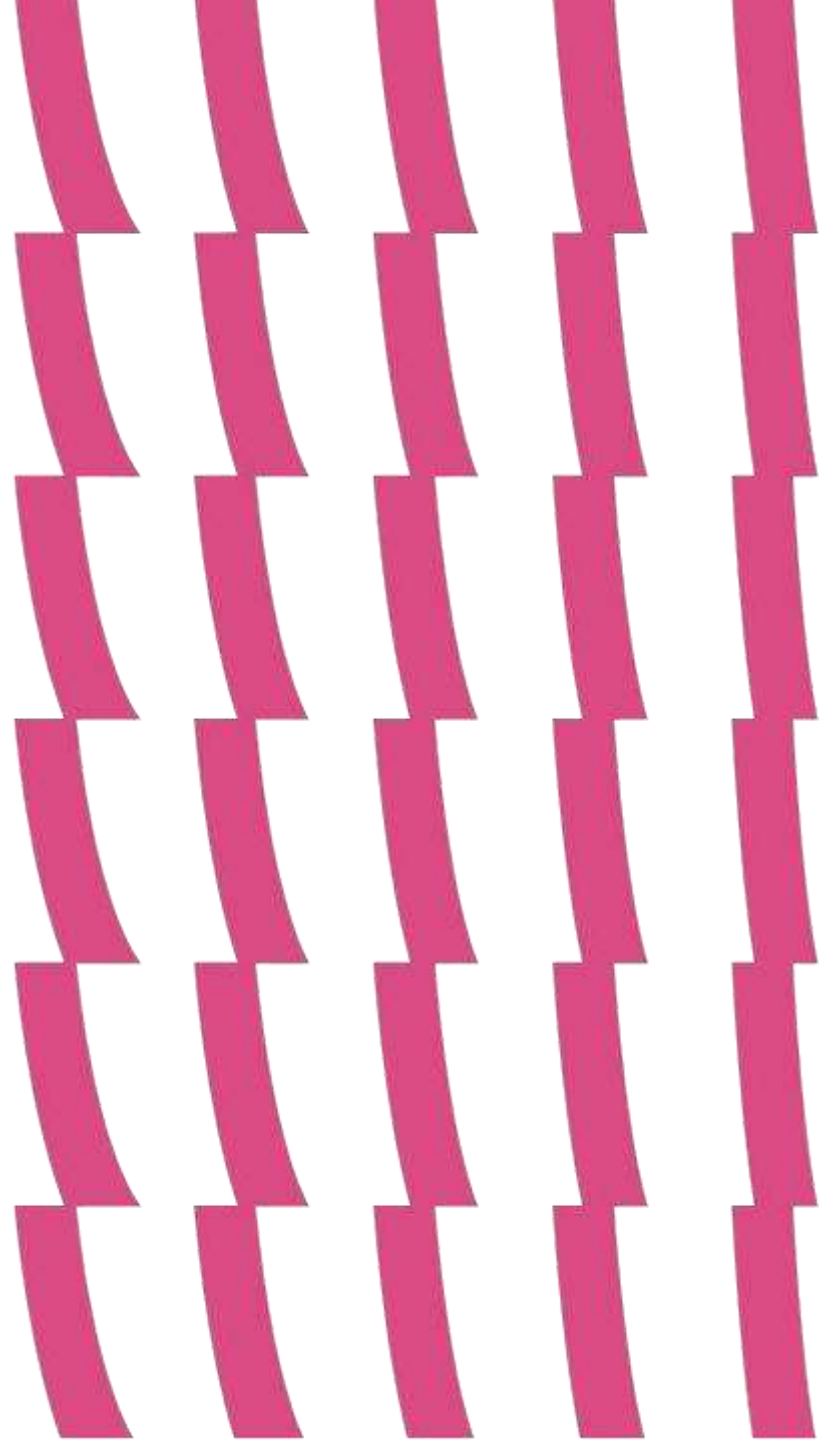
1. Нет “единой правды”
2. Чем больше и сложнее бизнес-область, которая должна быть загружена в хранилище, тем больше избыточность данных.
3. Сложно добавлять новые поля и новые сущности, сложно менять модель в случае изменений в бизнесе – в любой момент доработка может превратиться в «эпическую задачу».
4. Пожирание места в БД



Какую модель использовать? Depends on…

Требования заказчика.

1. Если требования заказчика носят более общий, стратегический характер, если необходимо показывать состояние бизнеса в целом, отслеживать взаимосвязи между бизнес-сущностями => DWH-I
2. Если поставлены конкретные цели, и нужны определенные отчеты по ограниченному множеству процессов и с ними связанных сущностей => DWH-K (Однако, тут нужно быть уверенным, что заказчик со временем не перейдет от конкретных требований к более масштабным).

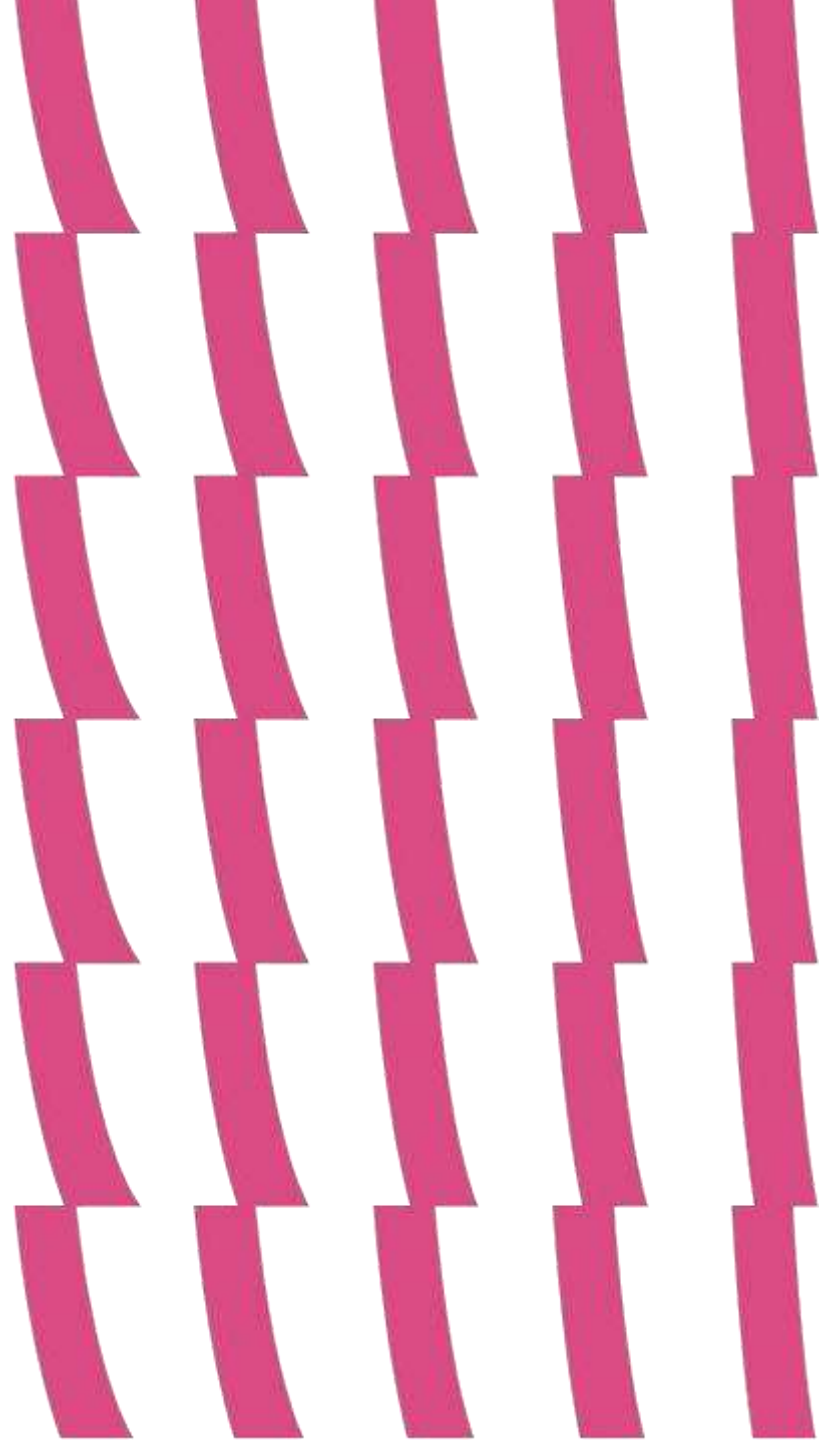


Выводы по лекции. Какую модель использовать?

Сроки. Нужно помнить, что на проектирование по Инмону требуется больше времени, чем на реализацию подхода Кимбалла. В условиях сжатых временных рамок => DWH-K

Выводы по лекции. Какую модель использовать?

Команда. Если заказчик может позволить себе команду высококвалифицированных (и иногда дорогостоящих) специалистов для доработки и поддержки хранилища => DWH-I



Выводы по лекции. Какую модель использовать?

Частота изменений.

В условиях развивающегося бизнеса, когда постоянно добавляются новые и изменяются старые сущности => DWH-I (в силу его гибкости - он предполагает высокую степень нормализации детального слоя)

Перспективы.

Если заказчик видит необходимость в долгосрочном развитии хранилища, понимает его цели и задачи => DWH-I Если заказчику необходимо получить конкретные отчеты => DWH-K

Спасибо
за внимание!