ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ
БАЗЫ ДАННЫХ
SQL

Доррер Михаил Георгиевич

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

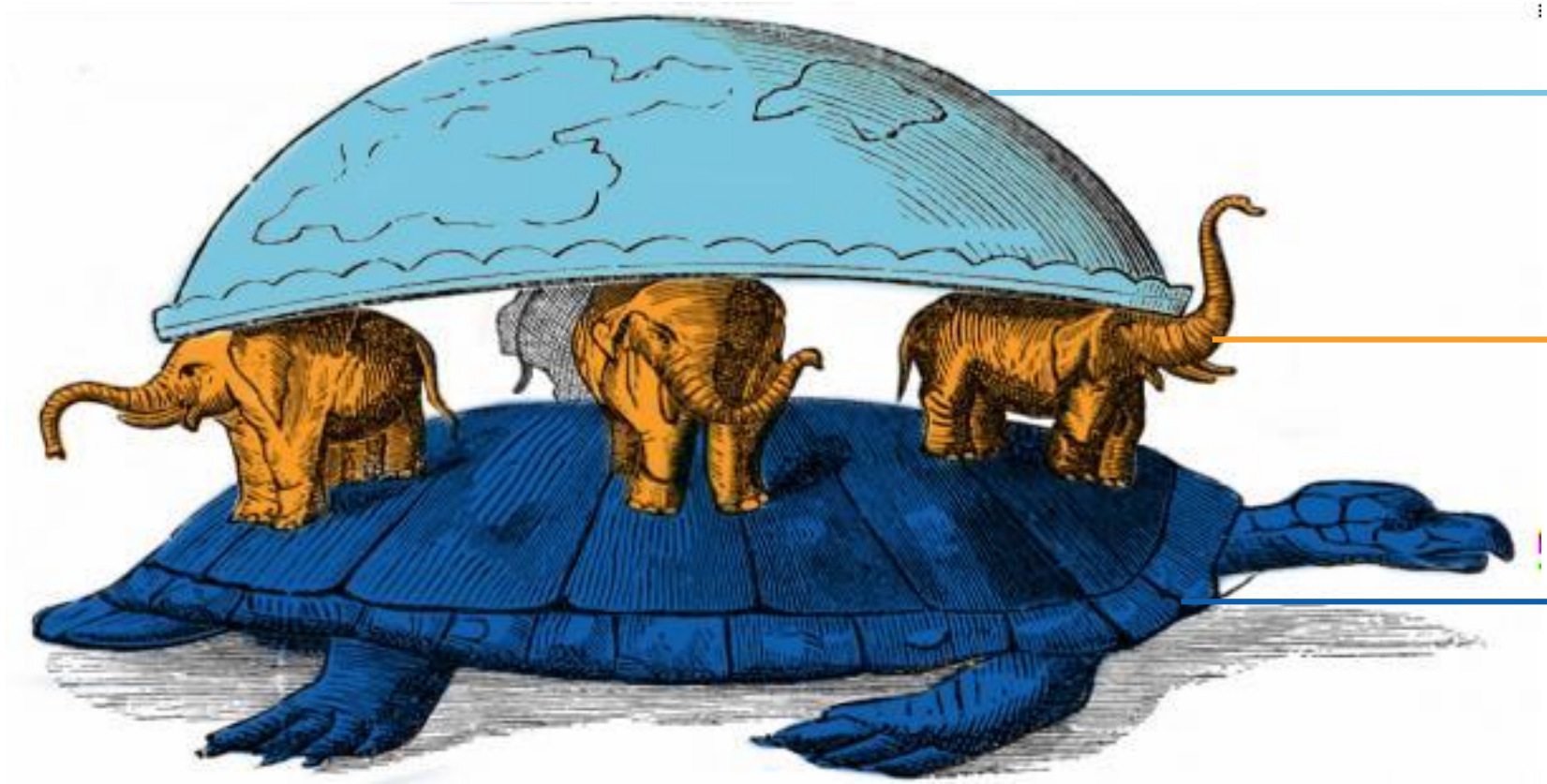
Содержание

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ (СХД)

- **Системы хранения данных –**
  - это программно-аппаратный комплекс. Основная задача - это хранение /передача больших массивов информации и предоставление к ним гарантированного доступа.
- **Структура СХД**
  - Устройство хранения
  - Инфраструктура доступа к устройствам
  - хранения
  - Подсистема резервирования и архивирования данных
  - ПО для управления хранением
  - Система управления и мониторинга

ПЛАТФОРМЫ ХРАНЕНИЯ ДАННЫХ



Потребители

Серверные

Распределенные

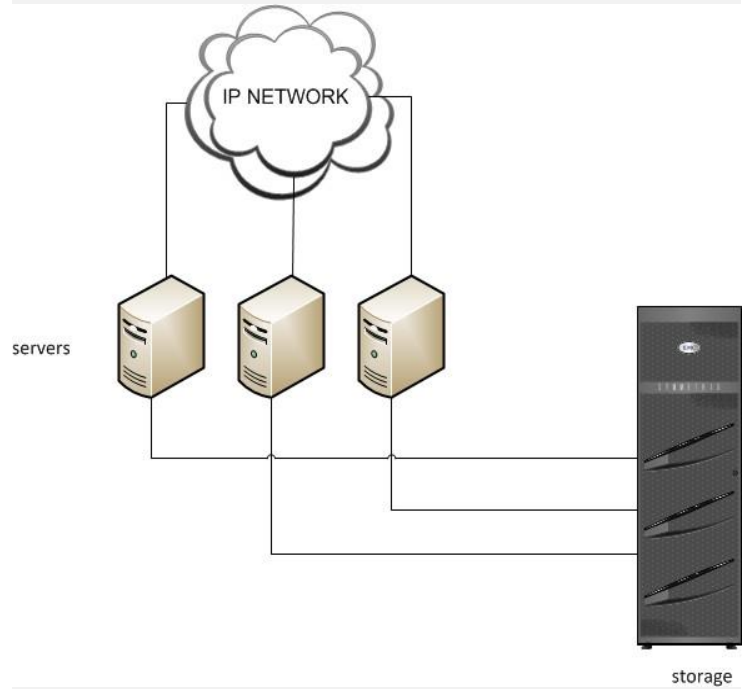
Облачные

Данные

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

DAS (Direct Attach Server)

- **DAS (Direct Attach Server) - DAS –**
  - система хранения, где физические диски подключаются напрямую к серверу или пользовательскому компьютеру.
- **Основные свойства:**
  - Диски подключены напрямую
  - Наименьшая цена по сравнению с
  - другими архитектурами
  - Возможно внешнее подключение через SCSI/FC
  - Наибольшая скорость доступа к данным



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

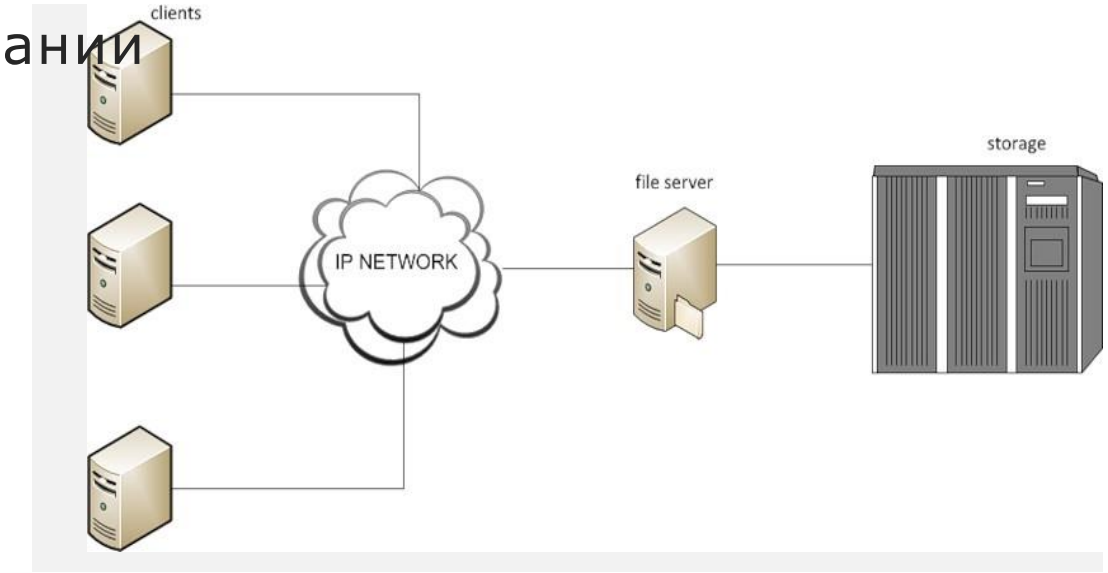
DAS – преимущества и недостатки

- **Плюсы**
  - Достаточно низкая стоимость. По сути эта СХД представляет собой дисковую корзину с жесткими дисками, вынесенную за пределы сервера.
  - Простота развертывания и администрирования.
  - Высокая скорость обмена между дисковым массивом и сервером.
- **Минусы**
  - Низкая надежность – при возникновении проблем в сети или аварии сервера данные становятся недоступны всем сразу.
  - Высокая латентность, обусловленная обработкой всех запросов одним сервером и использующимся транспортом (чаще всего – IP).
  - Высокая загрузка сети, часто определяющая пределы масштабируемости путём добавления клиентов.
  - Плохая управляемость – вся ёмкость доступна одному серверу, что снижает гибкость распределения данных.
  - Низкая утилизация ресурсов – трудно предсказать требуемые объёмы данных, у одних устройств DAS в организации может быть избыток ёмкости (дисков), у других её может не хватать – перераспределение часто невозможно или трудоёмко.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

NAS (Network Attached Storage)

- **NAS (Network Attached Storage) –**
  - система хранения, где накопитель присоединен к сети. При таком типе подключения используется только файловая передача данных.
  - Устройства NAS, часто называемые файлерами, состоят из единого головного устройства, выполняющего обработку данных и осуществляющего сетевое соединение цепочки дисков.
- **Основные свойства:**
  - Используется для файловой передачи
  - Просты в настройке и администрировании
  - Работают с большим пулом серверов
  - Большой объем памяти под кэш



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

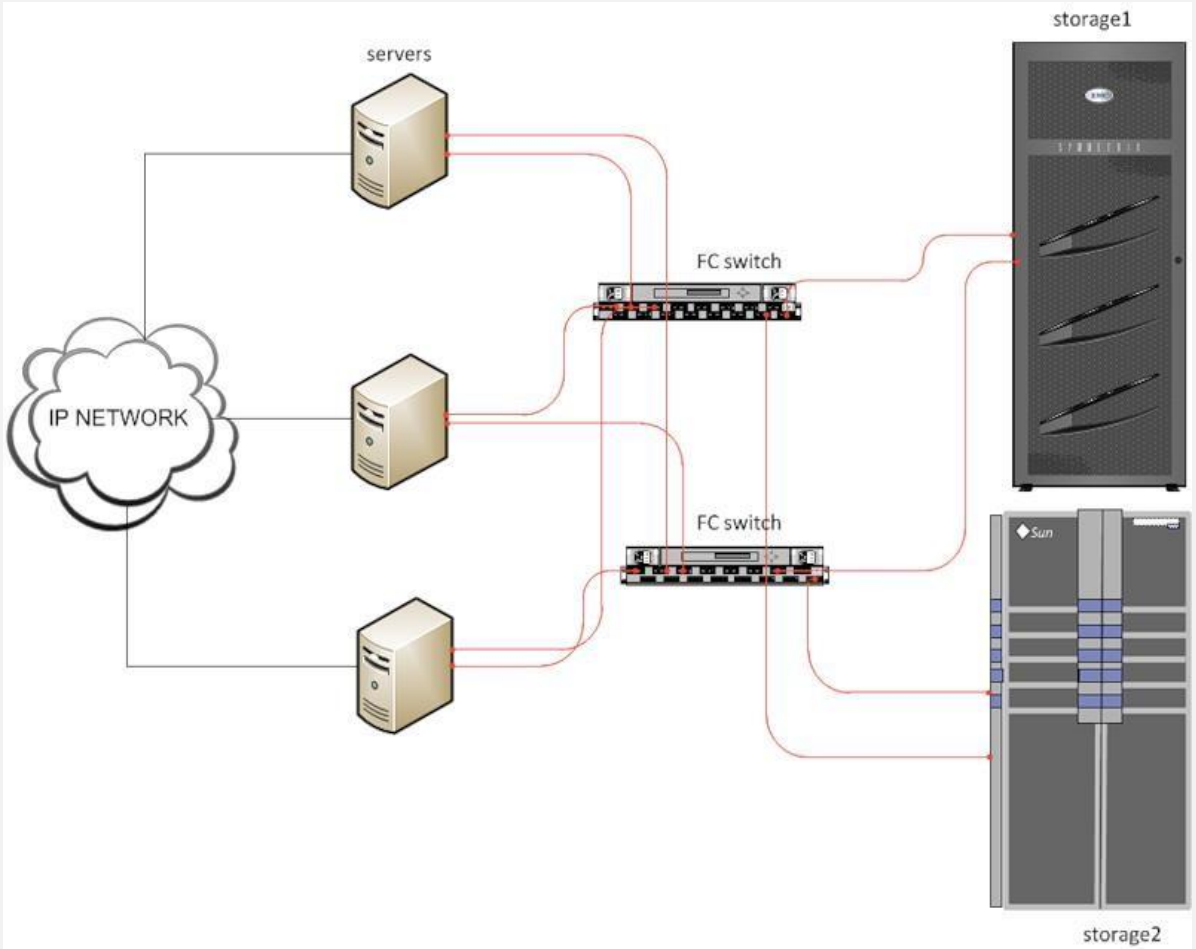
NAS – преимущества и недостатки

- **Плюсы**
  - Дешевизна и доступность его ресурсов не только для отдельных серверов, но и для любых компьютеров организации
  - Простота коллективного использования ресурсов
  - Простота развертывания и администрирования
  - Универсальность для клиентов (один сервер может обслуживать клиентов MS, Novell, Mac, Unix)
- **Минусы**
  - Доступ к информации через протоколы «сетевых файловых систем» зачастую медленнее, чем как к локальному диску.
  - Большинство недорогих NAS-серверов не позволяют обеспечить скоростной и гибкий метод доступа к данным на уровне блоков, присущих SAN системам, а не на уровне файлов.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

SAN (Storage Area Network)

- **SAN (Storage Area Network)**
  - это сеть хранения данных. Обычно при такой конфигурации используется блочный тип хранения данных.
  - представляет собой архитектурное решение для подключения внешних устройств хранения данных, таких как дисковые массивы, ленточные библиотеки, оптические накопители к серверам таким образом, чтобы операционная система распознала подключённые ресурсы, как локальные.
- **Основные свойства:**
  - Основной компонент - сеть
  - Доступ к данным на блочном уровне
  - Данные передаются по FC/Fast GB
  - Горизонтальное масштабирование
  - Множество путей получения данных



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

SAN – преимущества и недостатки

- **Плюсы**
  - Дешевизна и доступность его ресурсов не только для отдельных серверов, но и для любых компьютеров организации
  - Простота коллективного использования ресурсов
  - Простота развертывания и администрирования
  - Универсальность для клиентов (один сервер может обслуживать клиентов MS, Novell, Mac, Unix)
- **Минусы**
  - Доступ к информации через протоколы «сетевых файловых систем» зачастую медленнее, чем как к локальному диску.
  - Большинство недорогих NAS-серверов не позволяют обеспечить скоростной и гибкий метод доступа к данным на уровне блоков, присущих SAN системам, а не на уровне файлов.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

РАСПРЕДЕЛЕННЫЕ И ОБЛАЧНЫЕ ПЛАТФОРМЫ

- **Преимущества:**
  - Дешевый порог входа и эксплуатация (О)
  - Легкое и быстрое масштабирование
  - Принцип Pay-As-You-Go (О)
  - Не зависят от Географии (О)
  - Легкость управления и администрирования
  - Надежность и отказоустойчивость
  - Высокая скорость доступа к данным

hadoop

Dropbox

amazon S3

The Google File System

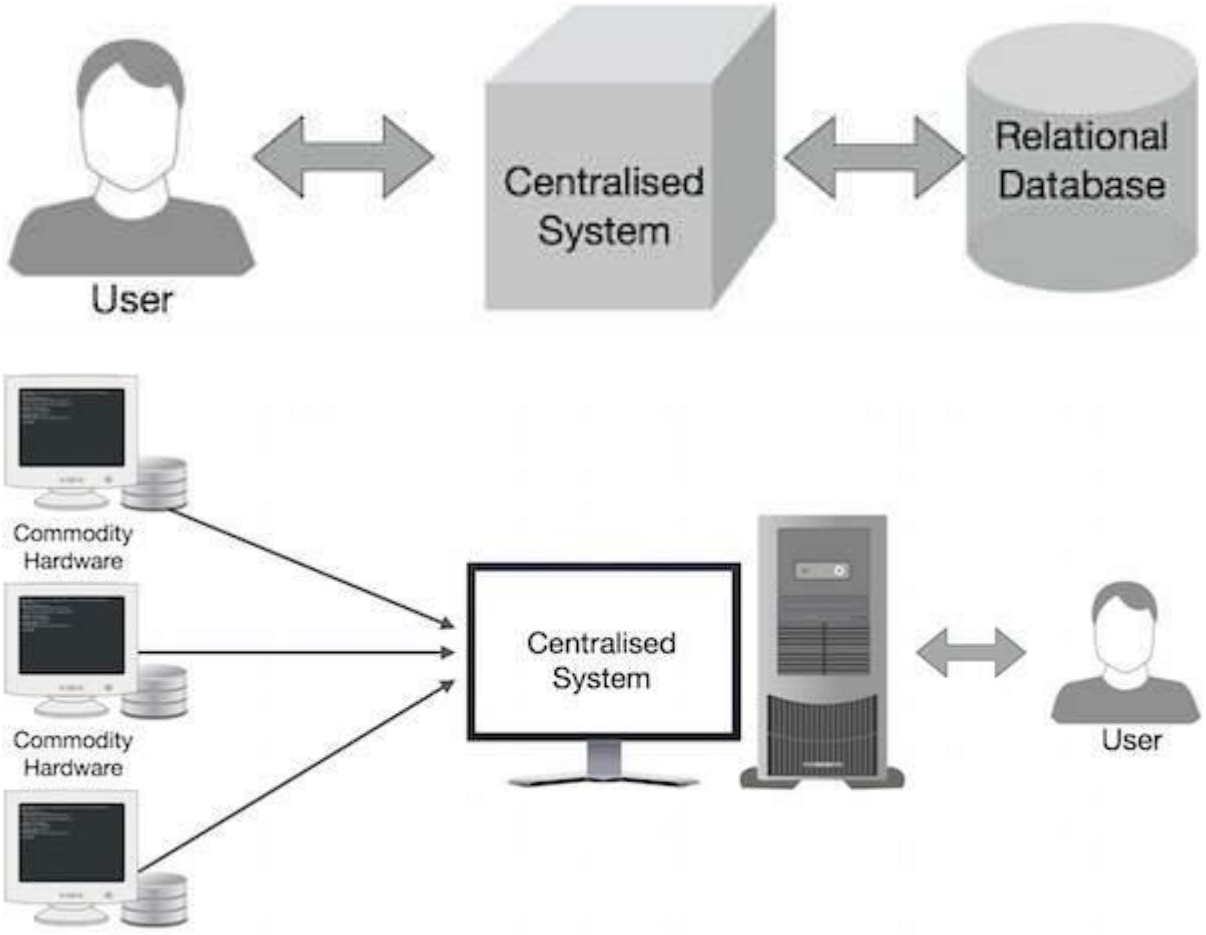
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Hadoop

- **Основан на парадигме MapReduce, предложенной компанией Google в 2004**
- **Изначально Hadoop был, в первую очередь, инструментом для хранения данных и запуска MapReduce-задач**
- **В настоящее время Hadoop представляет собой большой стек технологий, так или иначе связанных с обработкой больших данных (не только при помощи MapReduce).**
- **Основные (core) компоненты Hadoop:**
  - Hadoop Distributed File System (HDFS) – распределённая файловая система, позволяющая хранить информацию практически неограниченного объёма.
  - Hadoop YARN – фреймворк для управления ресурсами кластера и менеджмента задач, в том числе включает фреймворк MapReduce
  - Hadoop common

Идея MapReduce

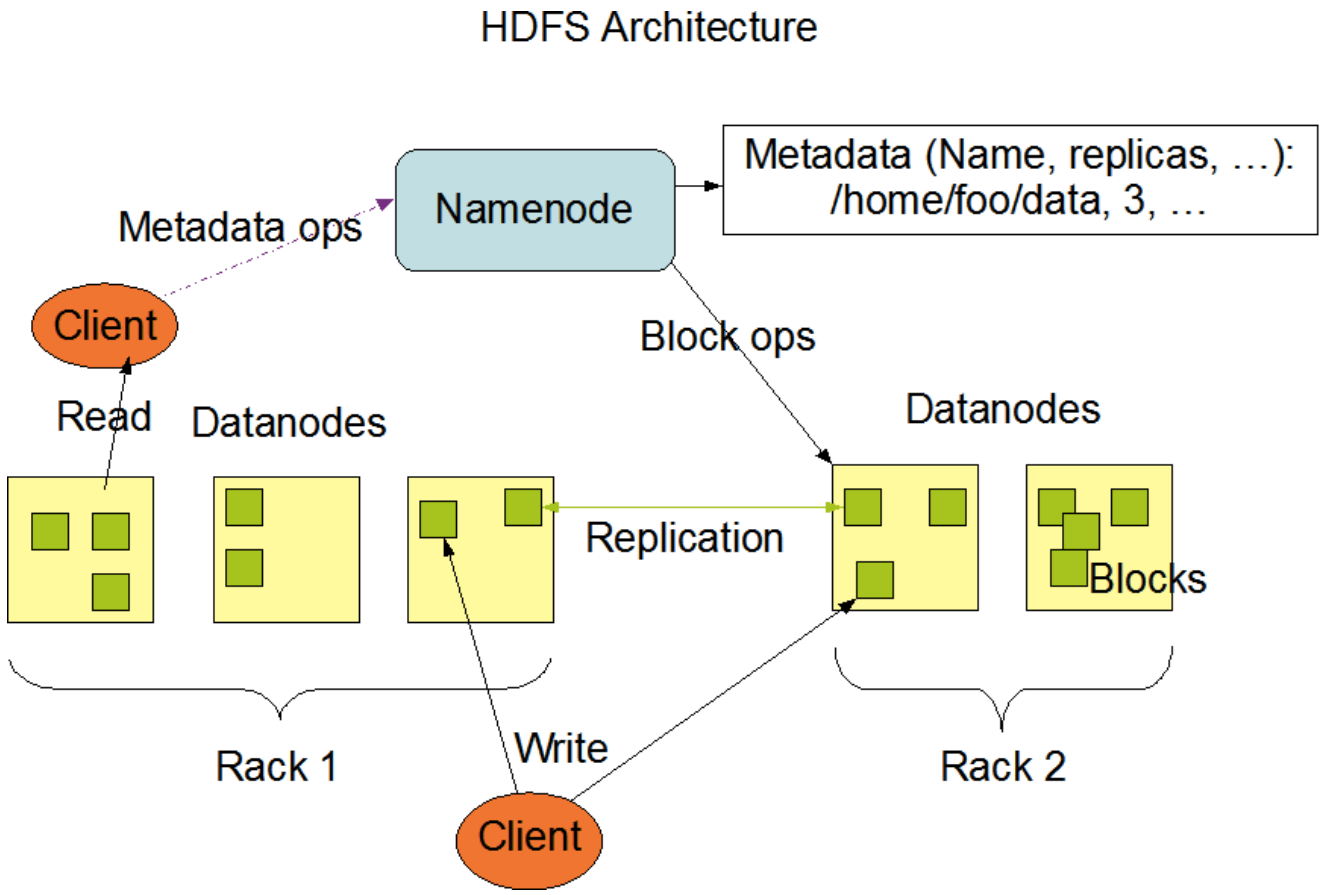
- Традиционная модель, безусловно, не подходит для обработки огромных объемов масштабируемых данных и не может быть размещена на стандартных серверах баз данных.
- Централизованная система создает слишком много узких мест при одновременной обработке нескольких файлов.
- Google решил эту проблему, используя алгоритм MapReduce. MapReduce делит задачу на маленькие части и назначает их многим компьютерам. Позже результаты собираются в одном месте и объединяются для формирования результирующего набора данных.



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Hadoop Distributed File System (HDFS)

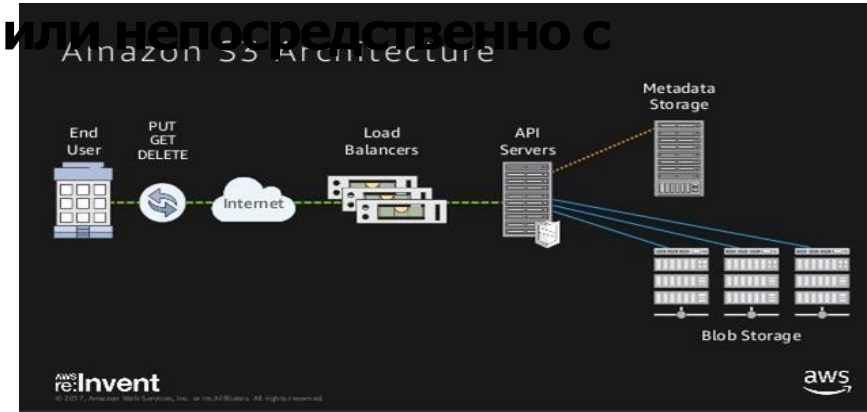
- **Hadoop Distributed File System**
  - Высокая надежность
  - Относительная дешевизна
  - Горизонтальное масштабирование
  - Вычисления переносятся к данным а не наоборот
  - Нет общих ресурсов, кроме NameNode
  - Простота доступа к данным



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

SIMPLE STORAGE SERVICE (S3)

- **AWS S3 – облачная платформа хранения данных от компании Amazon**
- **Содержит различные инструменты, которые позволяют организовать и контролировать для поддержки определенных сценариев использования, сокращения расходов, обеспечения безопасности и соблюдения законодательных требований.**
- **Данные хранятся как объекты в ресурсах, которые называют корзинами (бакетами), при этом размер одного объекта может составлять до 5 ТБ.**
- **Хранилище S3 позволяет**
  - добавлять теги метаданных в объекты,
  - перемещать и сохранять данные в классах хранилища S3,
  - настраивать и применять элементы управления доступом к данным, защищать данные от несанкционированного использования,
  - применять аналитику больших данных,
  - выполнять мониторинг данных на уровне объекта и корзины
  - просматривать статистику использования хранилищ и тенденции активности в своей организации.
- **Доступ к объектам можно получить через точки доступа S3 или непосредственно с помощью имени узла контейнера.**



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

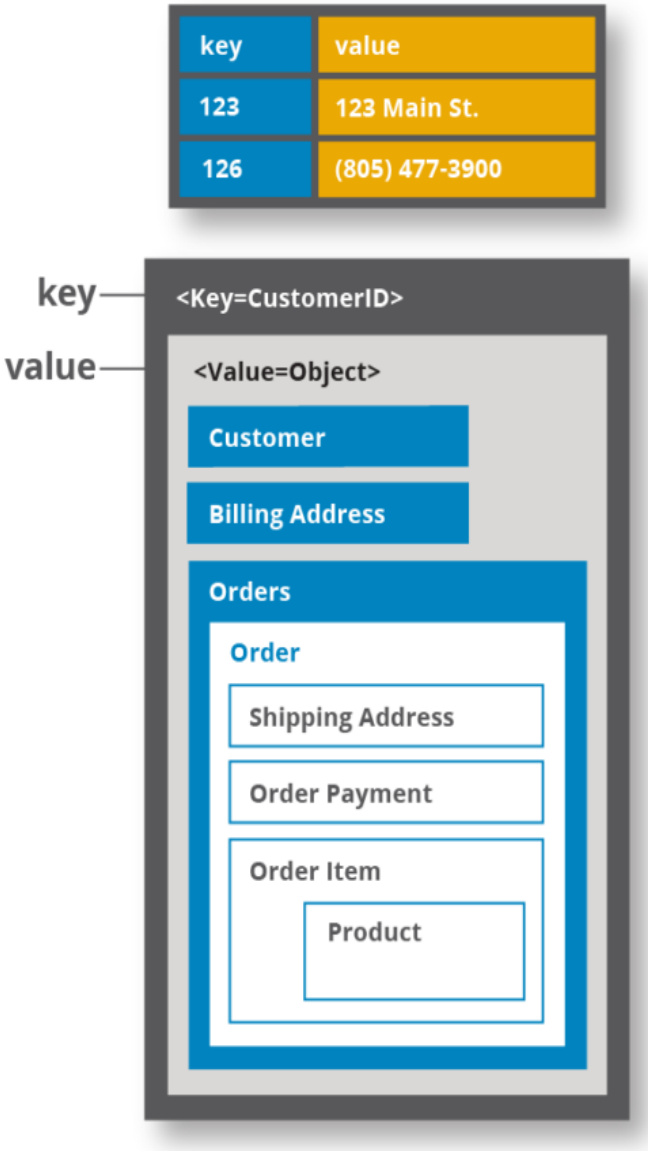
SIMPLE STORAGE SERVICE (S3)

- **AWS S3 облачная платформа хранения данных от компании Amazon**
  - Объектная модель
  - Данные хранятся в бакетах (каждый бакет уникален)
  - Надежность
  - Бесконечная емкость
  - Простота масштабированности и управления
  - Поддержка RESTful API
  - Дешевизна

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

KEY VALUE STORAGE

- **Key-value store, или Key-value DataBase**
  - представляет собой тип ПО для хранения и запросов к данным. Данные хранятся в виде набора уникальных ключей, с ассоциированными каждому ключу значениями. Этот принцип так же известен как «key-value пары». Значение может быть как простым объектом (строка, число), так и сложным – документ, вложенная таблица, вложенные key-value.



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ОСОБЕННОСТИ KEY-VALUE

Concurrency : В Key/Value Store параллелизм применим только к одиночному ключу. Часто модель параллельные запросы на чтение/запись часто разрешаются в рамках модели «согласованность в конечном счете» (eventually consistency) – через какой-то промежуток времени все запросы будут возвращать последнее обновленное значение.

Queries : Существует только один способ выполнения запросов в структуре key-value, - это запросы по ключу. Запрос по диапазону ключей в стандартном виде тоже не возможен. Однако запросы возможны посредством составных ключей с использованием префиксов/суффиксов.

Transactions : В большинстве реализаций транзакции невозможны (не гарантированы), доступ осуществляется по модели «согласованность в конечном счете»

Schema : Не имеют схемы, вся обработка остается на усмотрение программы потребителя.

Scaling up : Легко и быстро масштабируются за счет партиционирования ключей

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

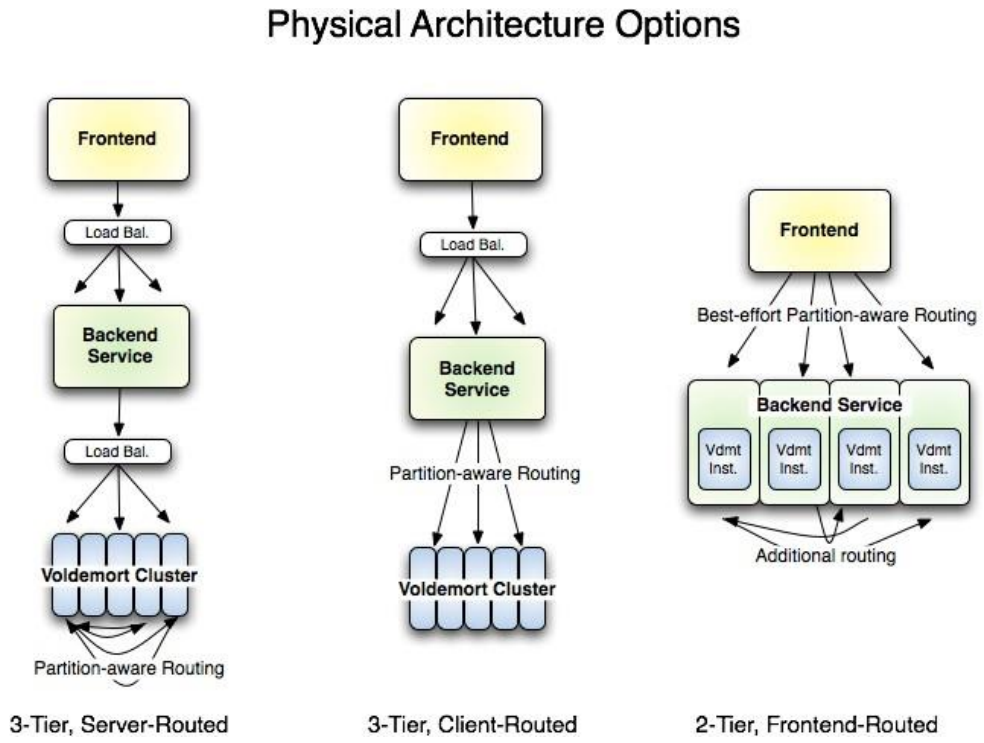
ОСНОВНЫЕ USE CASE

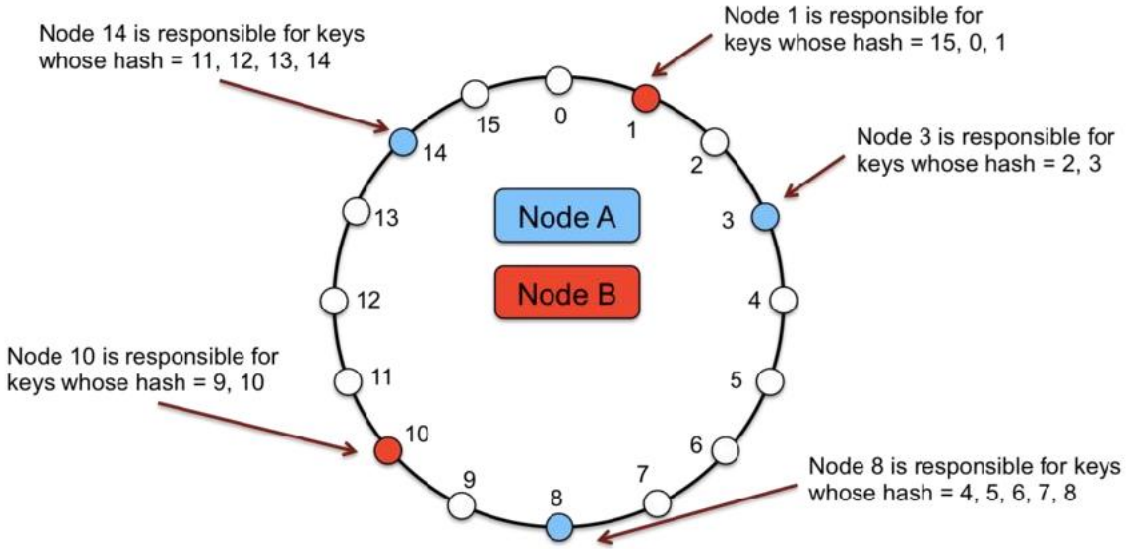
- Web Приложения – хранение пользовательский сессий, профилей. Ключом является пользователь, а value – как правило объект содержащий различную информацию. Такие запросы оптимизированы на быструю запись/чтение
- Системы Real-time рекомендаций и рекламные платформы, для быстрого добавления вновь посещенных сайтов/страниц и показа наиболее актуальных рекомендаций
- Большинство in-Memory баз данных имеют key-value структуру
- Расчет различных агрегационных показателей за разные периоды по большому количеству сущностей
- Организация быстрого доступа к тяжелому контенту (видео/фото)

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

РАСПРЕДЕЛЕННОЕ KEY-VALUE STORAGE

Согласованное хеширование: набор целых чисел от 0 до $2^{32}-1$, «закручивается» в кольцо. Каждому сервера из пула сопоставляем число на кольце. Ключ хэшируется в число в том же диапазоне, в качестве сервера для хранения ключа мы выбираем сервер в точке, ближайшей к точке ключа в направлении по часовой стрелке. Если сервер удаляется из пула или добавляется в пул, на оси появляется или исчезает точка сервера, в результате чего лишь часть ключей перемещается на другой сервер.





ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ОСНОВНЫЕ КОМПОНЕНТЫ БД

- **Диспетчер процессов**. Во многих БД имеется пул процессов/потоков, которыми нужно управлять. Причём в погоне за производительностью некоторые БД используют свои собственные потоки, а не предоставляемые ОС.
- **Диспетчер сети**. Пропускная способность сети имеет большое значение, особенно для
- распределённых БД.
- **Диспетчер файловой системы**. Первым «бутылочным горлышком» любой БД является производительность дисковой подсистемы. Поэтому очень важно иметь диспетчер, который идеально работает с файловой системой ОС или даже заменяет её.
- **Диспетчер памяти**. Чтобы не упереться в невысокую производительность дисковой подсистемы, нужно иметь много оперативной памяти. А значит, нужно эффективно ею управлять, что и делает данный диспетчер. Особенно когда у вас много одновременных запросов, использующих память.
- **Диспетчер безопасности**. Управляет аутентификацией и авторизацией пользователей
- **Диспетчер клиентов**.
- Управляет клиентскими соединениями.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

РЕЛЯЦИОННЫЕ БД

БД (База Данных) представляет собой совокупность информации, к которой можно легко получить доступ и модифицировать.

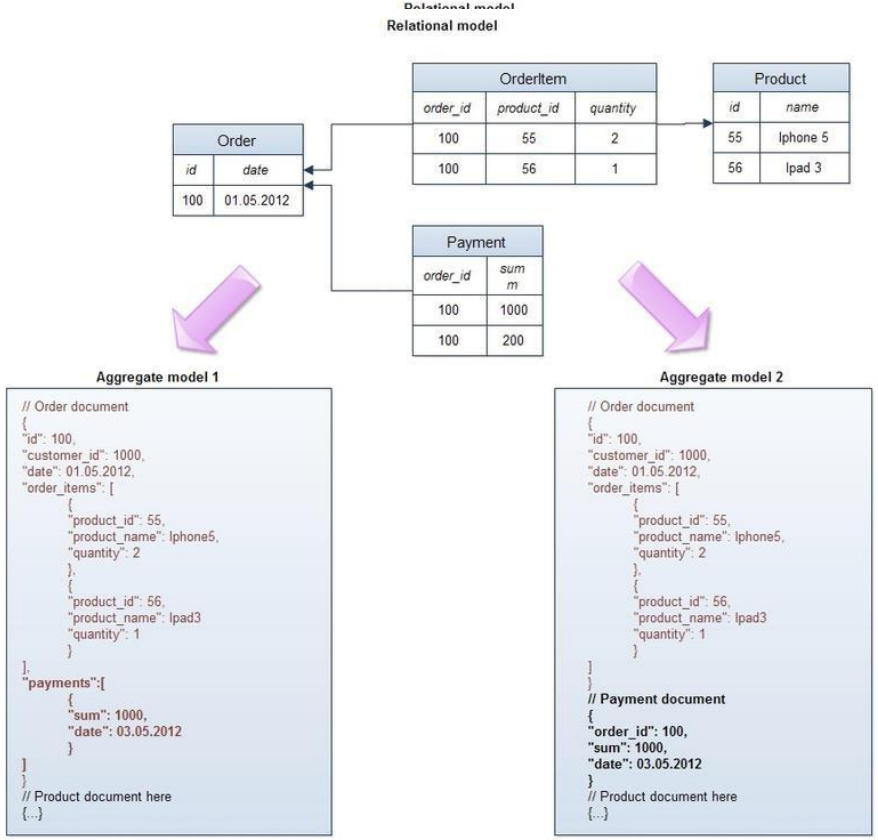
Особенности:

- Использование **транзакции** для обеспечения сохранности и связанности данных.
- Быстрая обработка данных вне зависимости от их объёма.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

НЕ РЕЛЯЦИОННЫЕ БД
NoSQL

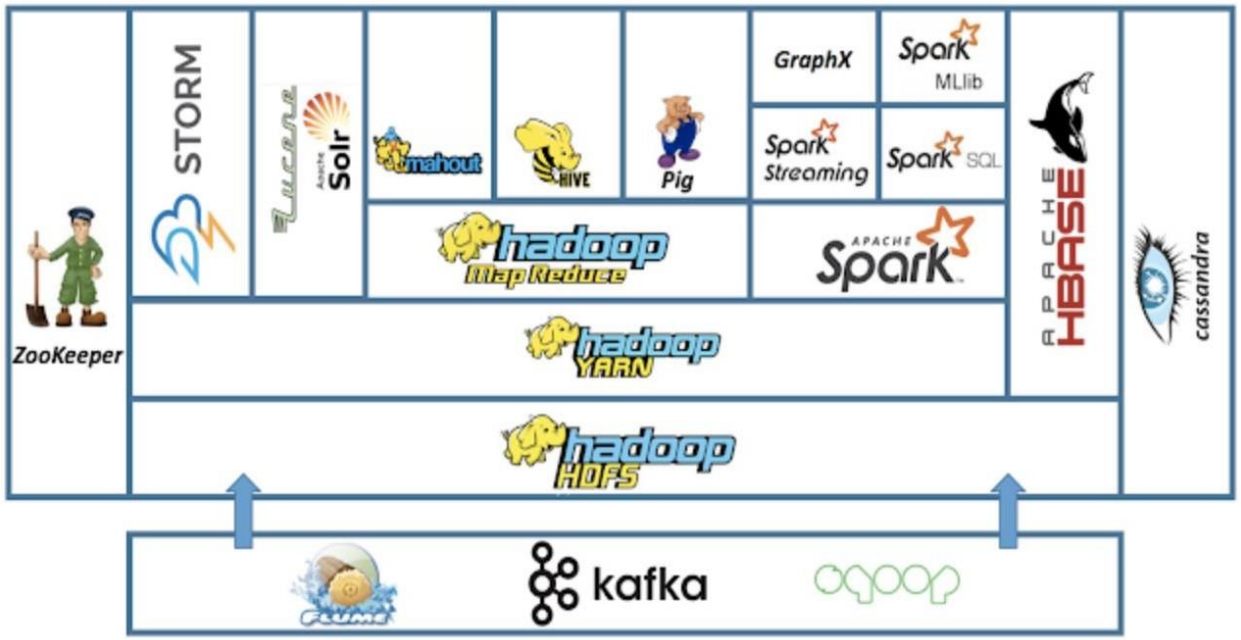
- Не используется SQL
- Неструктурированные (schemaless)
- Представление данных в виде объектов и их свойств (иерархические данные)
- Плохо поддерживают ACID.
- Распределенные системы, без совместно используемых ресурсов (share nothing).
- NoSQL базы в-основном бесплатны. (Opensource)



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

НЕ РЕЛЯЦИОННЫЕ БД
Hadoop

- Hadoop – Экосистема для работы с «Большими данными» (>200ТБ)
- HDFS – файловая система для хранения данных в Hadoop
- Mapreduce/Spark – движки для манипуляции с данными
- Поддержка SQL и NoSQL
- Потоковая обработка (Kafka/Spark Streaming)
- Машинное обучение
- Различные форматы хранения



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

РЕЛЯЦИОННЫЕ БД

БД (База Данных) представляет собой совокупность информации, к которой можно легко получить доступ и модифицировать.

Особенности:

- Использование **транзакции** для обеспечения сохранности и связанности данных.
- Быстрая обработка данных вне зависимости от их объёма.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СВОЙСТВА ТРАНЗАКЦИЙ

ACID-транзакция (Atomicity, Isolation, Durability, Consistency) — это элементарная операция, единица работы, которая удовлетворяет 4 условиям:

- **Атомарность (Atomicity).** Нет ничего «меньше» транзакции, никакой более мелкой операции. Даже если транзакция длится 10 часов. В случае неудачного выполнения транзакции система возвращается в состояние «до», то есть транзакция откатывается.
- **Изолированность (Isolation)**. Если в одно время выполняются две транзакции A и B, то их результат не должен зависеть от того, завершилась ли одна из них до, во время или после исполнения другой.
- **Надёжность (Durability).** Когда транзакция зафиксирована (commited), то есть успешно завершена, использовавшиеся ею данные остаются в БД вне зависимости от возможных происшествий (ошибки, падения).
- **Консистентность (согласованность) (Consistency).** В БД записываются только валидные данные (с точки зрения реляционных и функциональных связей). Консистентность зависит от атомарности и изолированности.

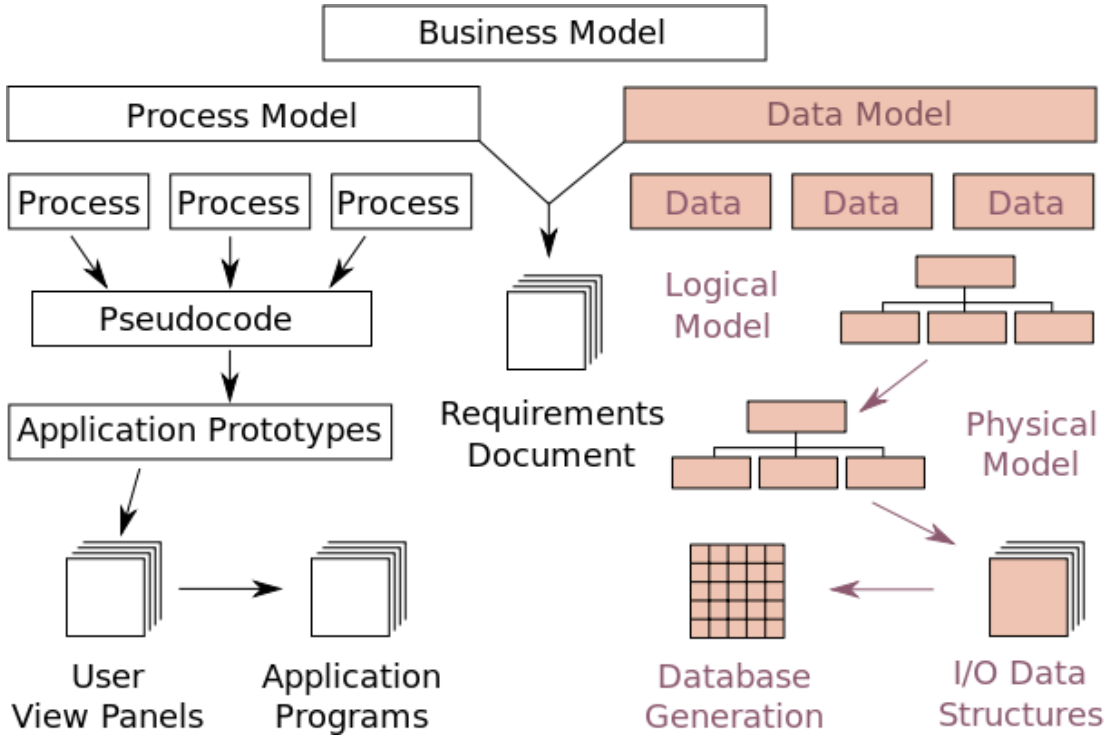
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

МОДЕЛИРОВАНИЕ ДАННЫХ

Моделирование данных используется для анализа и определения требований к данным для организации бизнес процесса, в задаче проектирования информационных систем.

Для описиания модели данных используются специальные языки:

- Entity–relationship diagram (ERD)
- UML



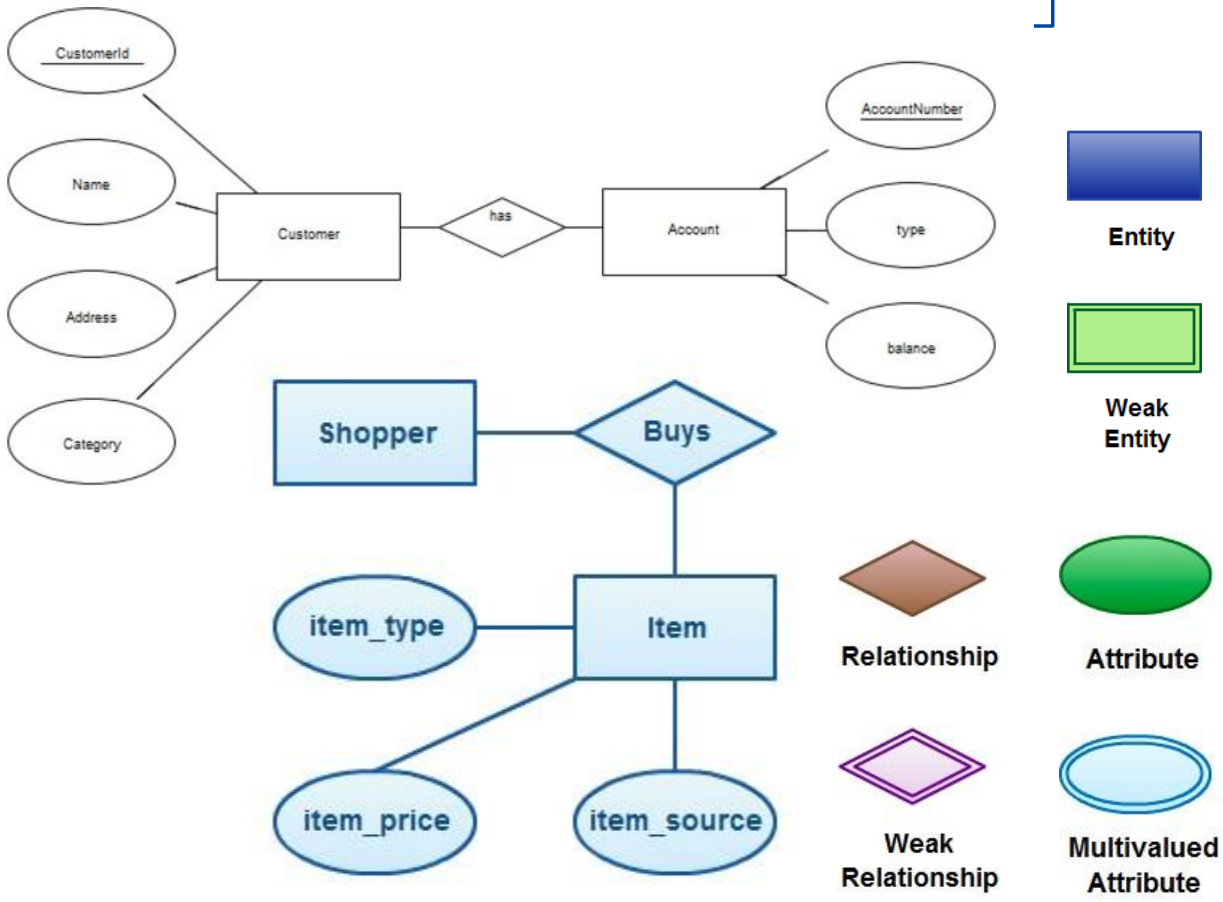
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ENTITY RELATION DIAGRAM

Entity Relationship Diagram (ERD) представляет собой визуальное представление различных сущностей в системах и как они относятся друг к другу.

Основные Элементы Диаграмы

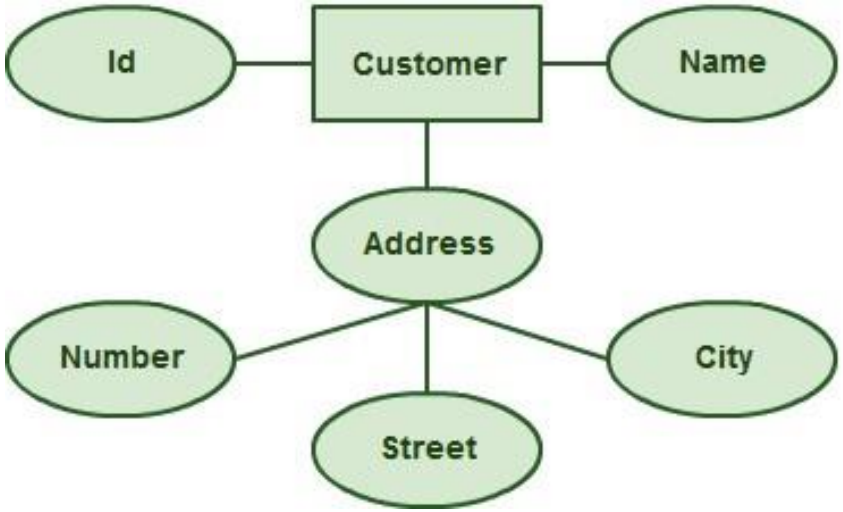
- Entity: независимый индивидуальный элемент(сущность) системы выражающий объект (покупатель, учиталь, дом) или концепцию (транзакции, отзывы).
- Entity type: категория определяющая одну группу сущностей.
- Relationship: Отражает зависимость между Entity (сущностями) системы.
- Attribute: описывает Entity или Relationship , представляет собой отлельную часть информации



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

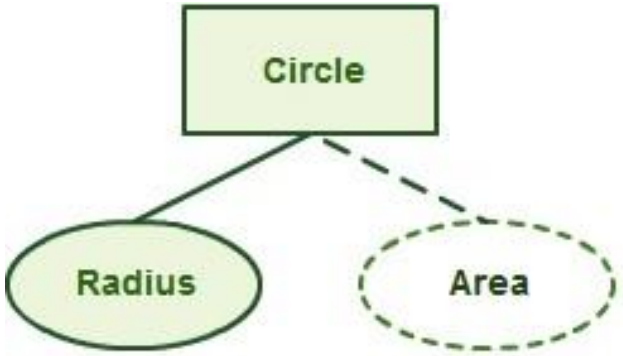
ENTITY RELATION DIAGRAM: ATTRIBUTE

Атрибут представляет собой свойства, особенность или характеристику Entity, Relationship или другого атрибута.
Атрибут так же может иметь вложенные атрибуты (составной атрибут): “customer address” имеет вложенные атрибуты: number, street, city.



Множественные атрибуты:

Производные атрибуты:



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

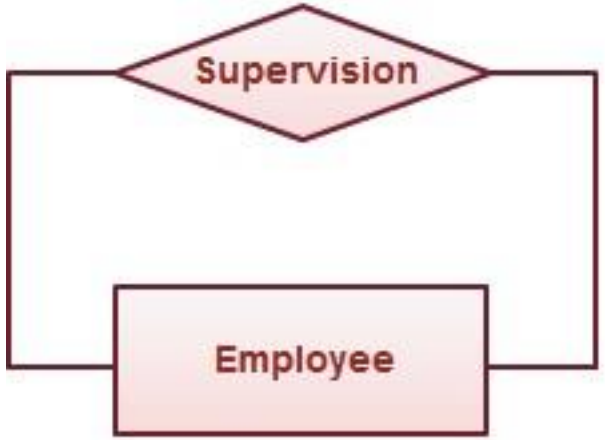
ENTITY RELATION DIAGRAM: RELATIONSHIP

Relationship описывает взаимодействие между Entity, например сущность “Carpenter” (столяр) может относиться к сущности “table” (стол) по средством relationship “makes” (делает). Relationships обозначаются с помощью значка ромб и содержат глагол определяющий это отношение.

Тип отношения (cardinality):

one2many, one2one, many2many

Рекурсивные отношения:

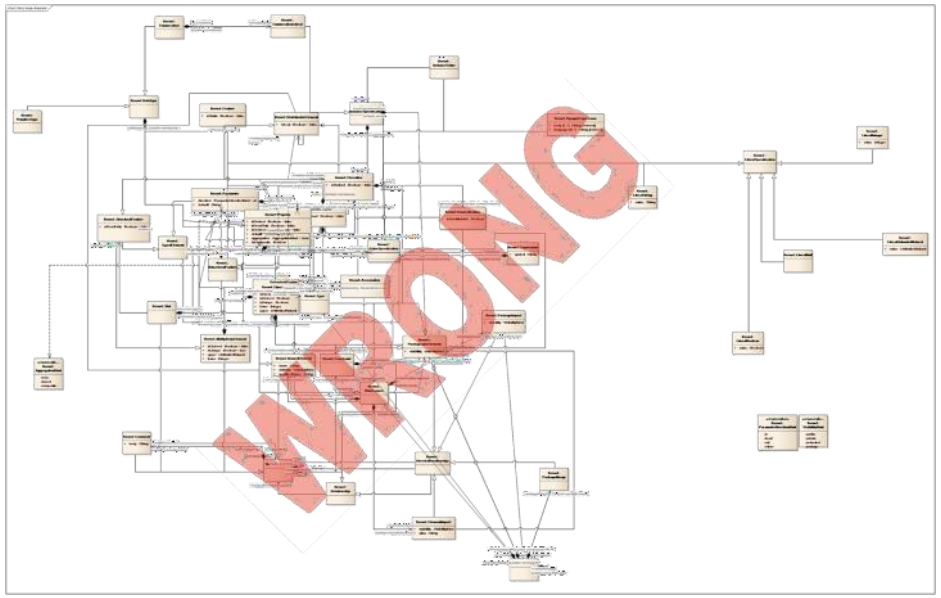


ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ENTITY RELATION DIAGRAM: RELATIONSHIP

Пререквизиты для создания диаграммы:

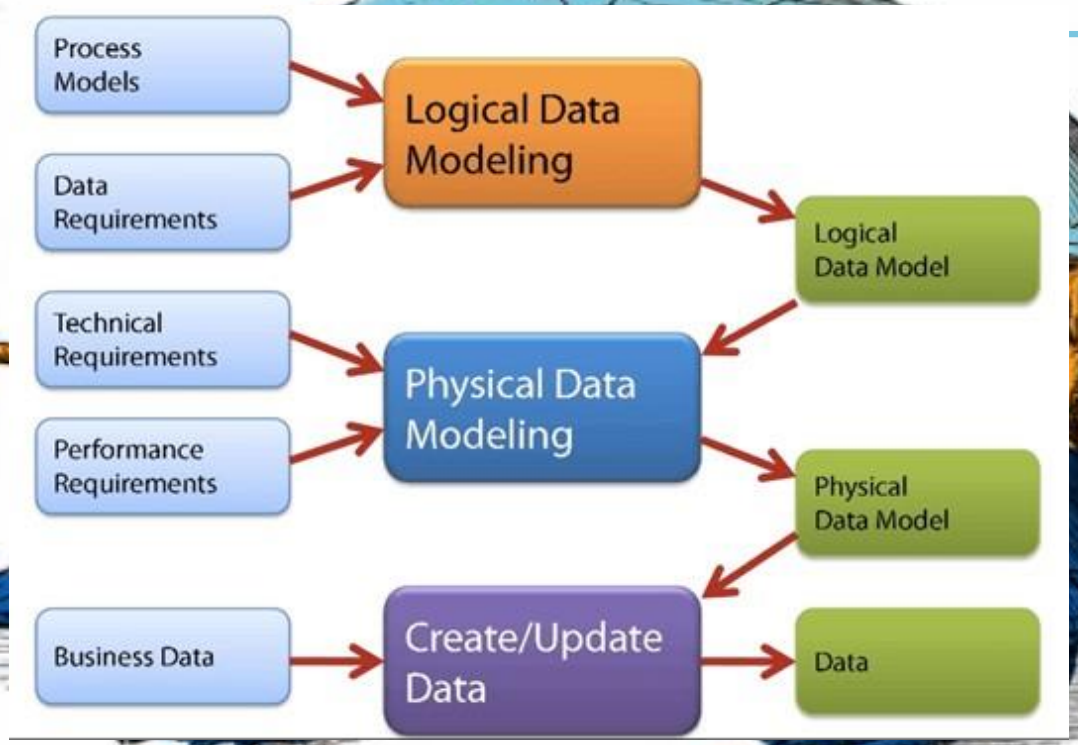
- Определить все entities в информационной системе. Entity должно быть представлено единожды в одной диаграмме (быть уникальным)
- Связать entity друг с другом с помощью relationships .
- Добавить attribute для каждого Entity. Имена атрибутов должны определять их сущность



Советы:

- Указывайте четкие и корректные имена для каждой сущности (простые и понятные всем термины предпочтительней технических терминов).
- Имена должны представлять собой имена существительные.
- Используйте прилагательные для разделения Entity входящих в один класс (part-time employee и full-time employee).
- Имена attributes должны быть понятные, уникальные, не привязаны к конкретной системе.
- Удаляйте все нечеткие, избыточные или ненужные relationship между entities.
- НИКОГДА не соединяйте один relationship с другим напрямую.
- Используйте цвета в диаграмме чтобы объединить entity относящиеся к одному классу или подчеркнуть ключевые области.

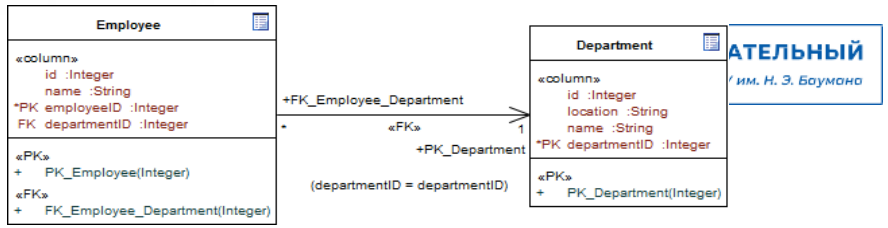
3 УРОВНЯ МОДЕЛИ ДАННЫХ



Концептуальный: Этот уровень определяет **ЧТО** будет содержать модель данных.
(Организация определение объема и бизнес концепта/правил для модели данных).

Логический: Определяет **КАК** информационная система будет внедрена без оглядки на конкретные требования баз данных или языка программирования. (основная цель это определить базовые типы данных и связей между ними)

Физический: Определяет **КАК** информационная система будет реализована с использованием конкретной БД или языка программирования.

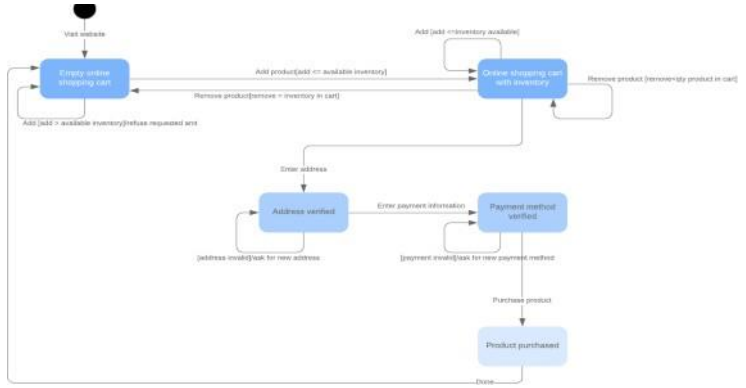


ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

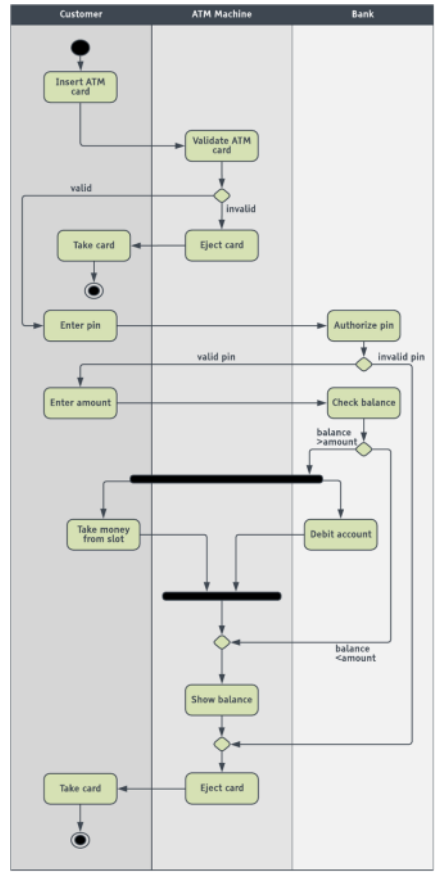
UML: ПОВЕДЕНЧЕСКИЕ ДИАГРАММЫ

Поведенческая UML диаграмма показывает как система себя ведет и взаимодействует внутри себя, с пользователями и внешними системами.

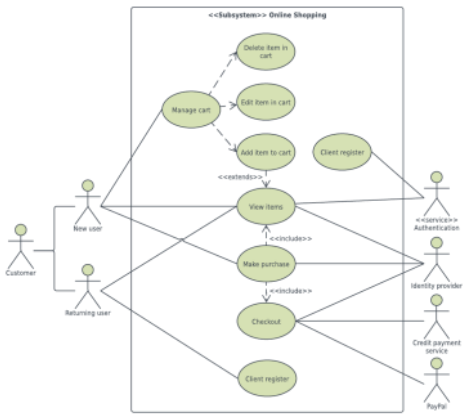
State UML



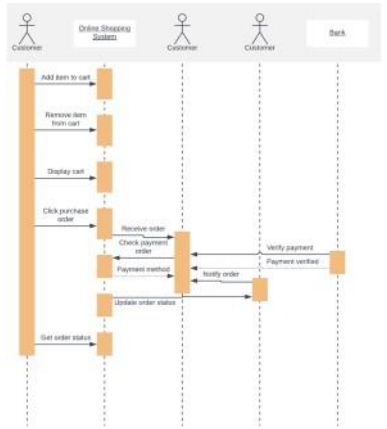
Activity UML



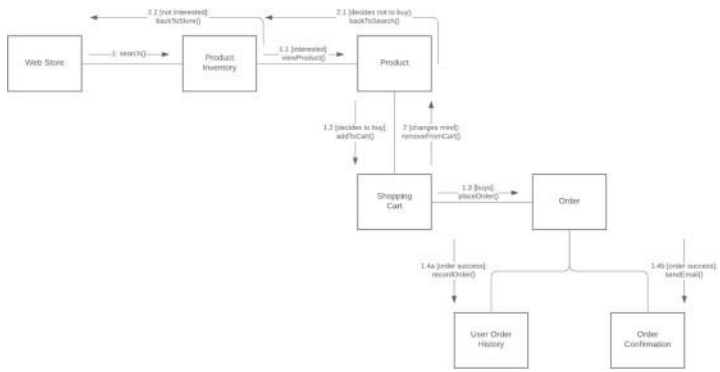
Use Case UML



Sequence UML



Communication UML

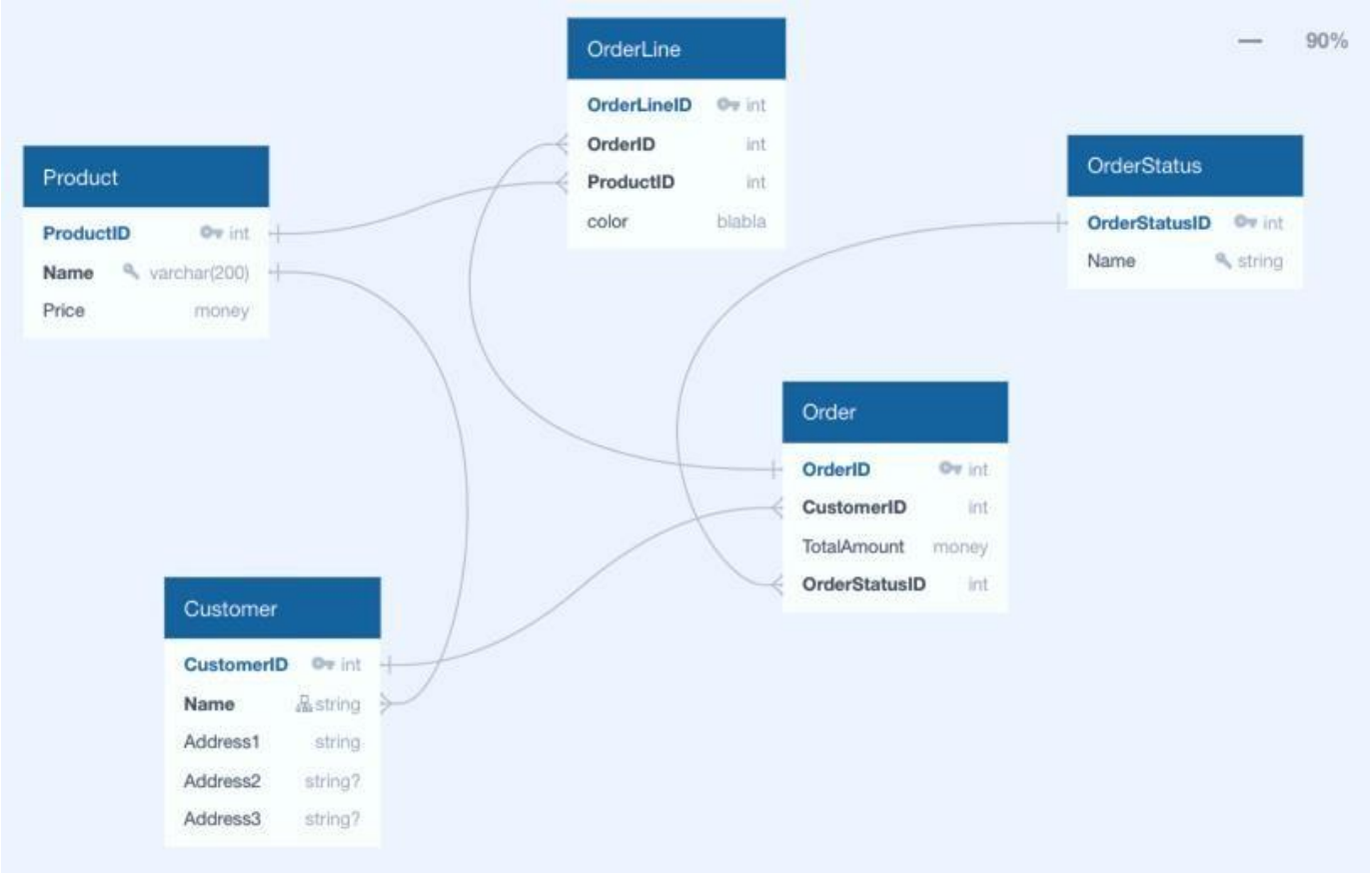


ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СХЕМА ДАННЫХ

Схема данных (схема БД) — Описывает структуру базы данных или отдельного отношения, включает в себя описание типов данных (атрибутов), ограничений и основных ключей.

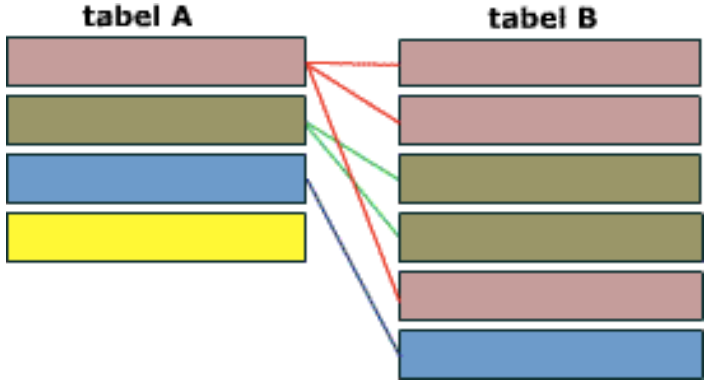
Схема так же описывает взаимодействие отношений в БД (таблиц) друг с другом.



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СВЯЗИ: ONE2MANY

Одна запись в таблице **A** может быть связана с **0**, **1**или **множеством записей** в таблице B.



Чтобы опознать ответьте на 2 вопроса

1) Сколько объектов и B могут относится к объекту A?

2)Сколько объектов из A могут относиться к объекту из B?

Один ко Многим (one2many)

на первый вопрос ответ – **множество**,

на второй – **один** (или ни одного)

Примеры:

- Машина и ее части. Каждая часть машины единовременно принадлежит только одной машине, но машина может иметь множество частей.
- Дома и улицы. На улице может быть несколько домов, но каждый дом принадлежит только одной улице.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СВЯЗИ: MANY2MANY

Связь, при которой **множественным записям** из одной таблицы (A) могут соответствовать **множественные записи** из другой (B).

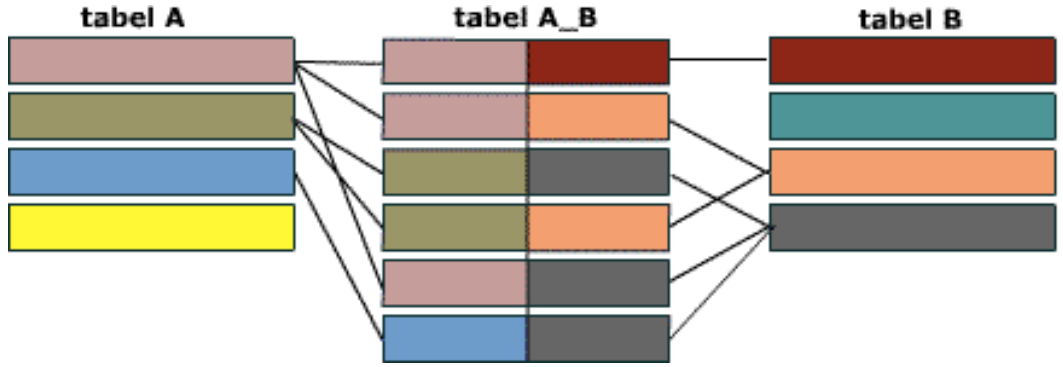


Table beer

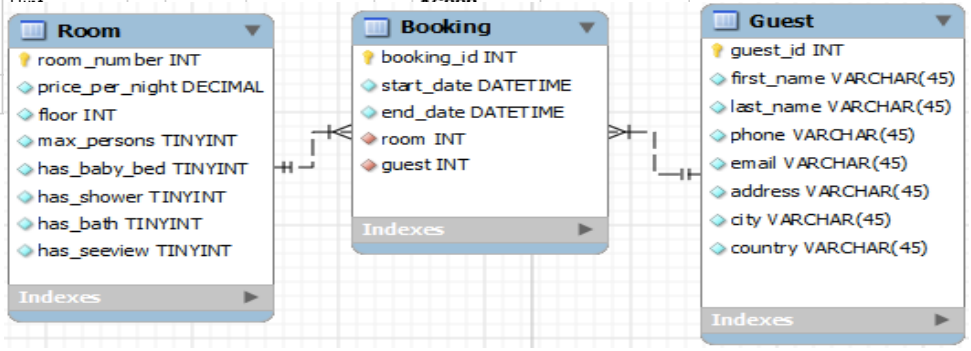
beer_id	name
157	Gentse Tripel
158	Uilenspiegel
146	Duvel
123	Leffe
222	Brugse Tripel
160	
163	

Table beer_distributor

beer_id	distributor_id
157	AC001
157	AB899
157	AC009
158	AC009
163	AC009
160	AB999
163	AB999

Table distributor

distributor_id	name
AB999	De vrolijke drinker
AC001	Horeca Import NL
AC002	van Gent bierimport
AB899	Jansen Horeca
	De

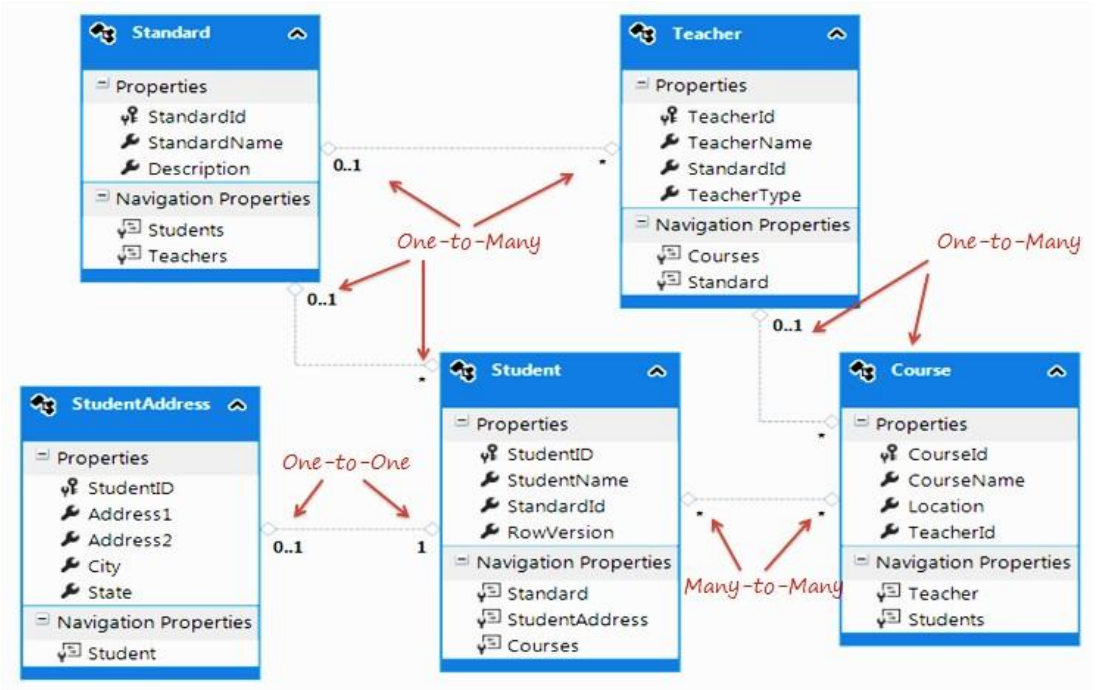


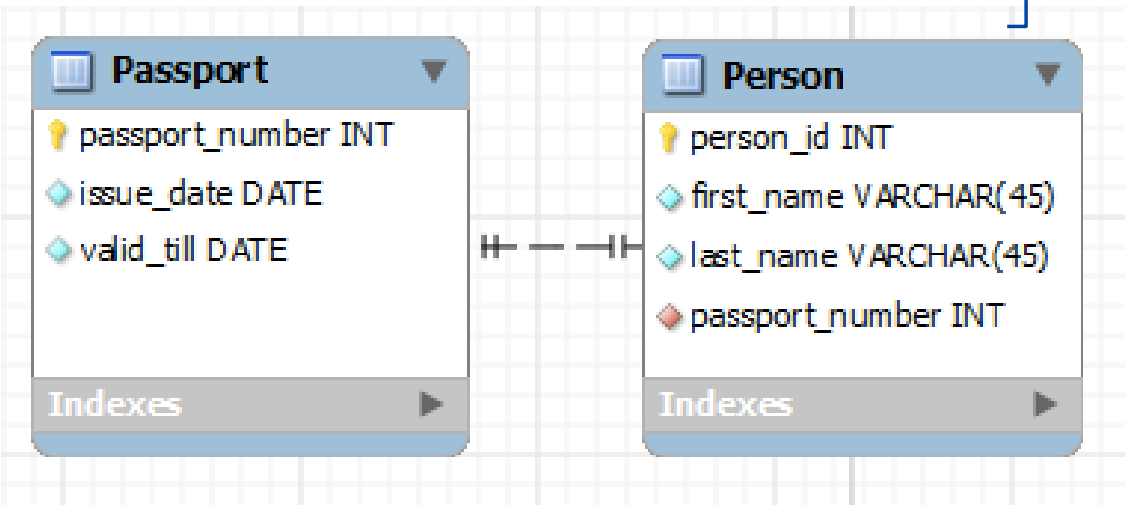
Две таблицы –“источника” и одна соединительная таблица. Первичный ключ соединительной таблицы A_B –**составной**. Она состоит из двух полей, двух внешних ключей, которые ссылаются на первичные ключи таблиц A и B. Все первичные ключи должны быть уникальными. Это подразумевает и то, что комбинация полей A и B должна быть уникальной в таблице A_B

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

СВЯЗИ: ONE2ONE

Связь, при которой **каждый** блок сущности А может быть ассоциирован **с 0 или 1** блоком сущности В.

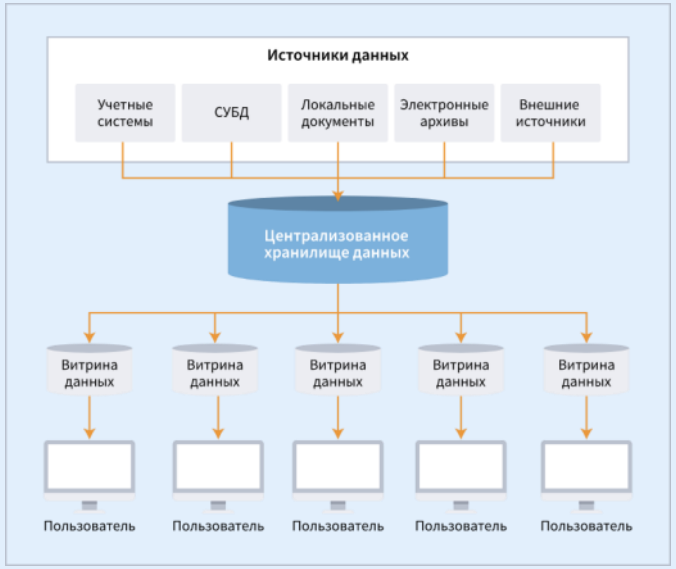




Пример:
Люди и их паспорта. Каждый человек в стране имеет только один действующий паспорт и каждый паспорт принадлежит только одному человеку.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВИТРИНА ДАННЫХ



Витрина Данных (Data Mart) - относительно небольшое хранилище или же его часть, предназначенную для применения конкретным подразделением организации и/или определенной группой пользователей.

Достоинства:

- Представление аналитикам только той информации, которая действительно нужна для определенного рабочего задания, профиля служебной деятельности.
- Максимальная приближенность целевой части хранилища данных к конкретному пользователю.
- Содержание тематических подмножеств заранее агрегированных специалистами данных, которые в дальнейшем проще настраивать и проектировать.
- Для реализации витрины данных (хранилища данных специализированного типа) не требуется вычислительная техника большой мощности.

Недостатки:

- Реализация информационной территориально распределенной системы, чья избыточность слабо контролируется.
- Не предполагается методик, способов, которые могли бы обеспечить целостность и непротиворечивость хранящейся в витрине данных (базе данных узкоспециальной) информации.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВИТРИНА ДАННЫХ VS ХРАНИЛИЩЕ ДАННЫХ 1/2

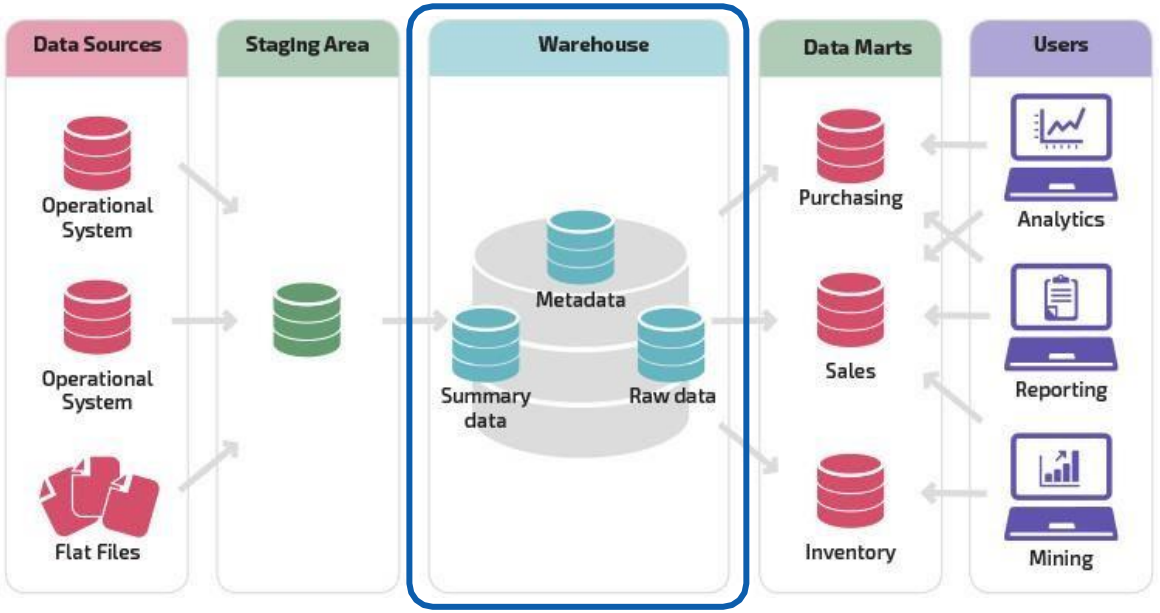
Хранилище Данных

Фокус: Промышленный репозиторий различных источников данных.

Источники данных: Множество внутренних и внешних источников данных из различных областей организации.

Размер: Начиная от 100 GB и до Петабайт

Нормализация: Современные хранилища слабо нормализованы для обеспечения наибольшей производительности.



Типы принимаемых решений: На основе данных из хранилища принимаются стратегические решения влияющие на всю организацию в целом.

Стоимость: Часто более $100,000.

Время развертывания и установки: Не менее года, при использовании собственной инфраструктуры

Типы Данных: сырые данные, справочники, транзакционные данные

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВИТРИНА ДАННЫХ VS ХРАНИЛИЩЕ ДАННЫХ 1/2

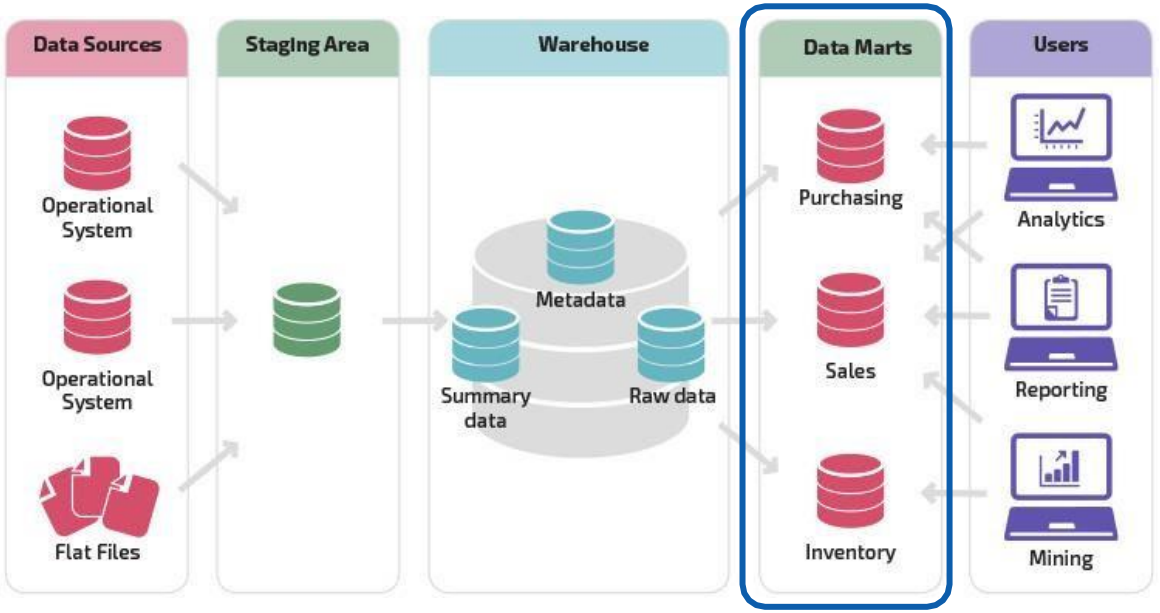
Витрина Данных

Фокус: Конкретный субъект в деятельности организации (например отдел продаж)

Источники Данных: Ограниченное кол-во источников данных связанных с бизнес потребителем.

Размер: Менее100 GB

Нормализация: Часто строго нормализованы, но позволяют гибко перестраивать отношения



Типы принимаемых решений: Тактические решения, затрагивающие в основном деятельность конкретного подразделения

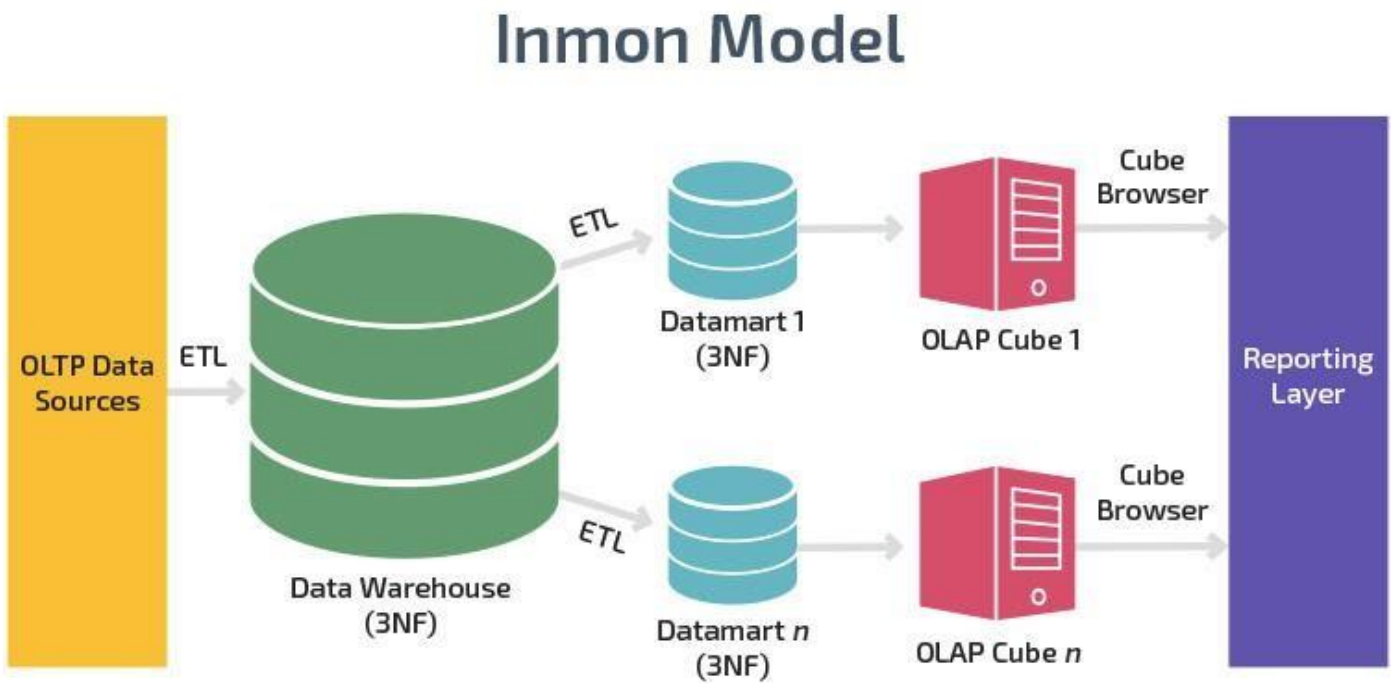
Стоимость: в районе $10,000 и выше

Время развертывания: 3-6 месяцев

Типы Данных: Аггрегаты , KPI и справочники

ИММОН VS КИМБАЛЛ 1/2

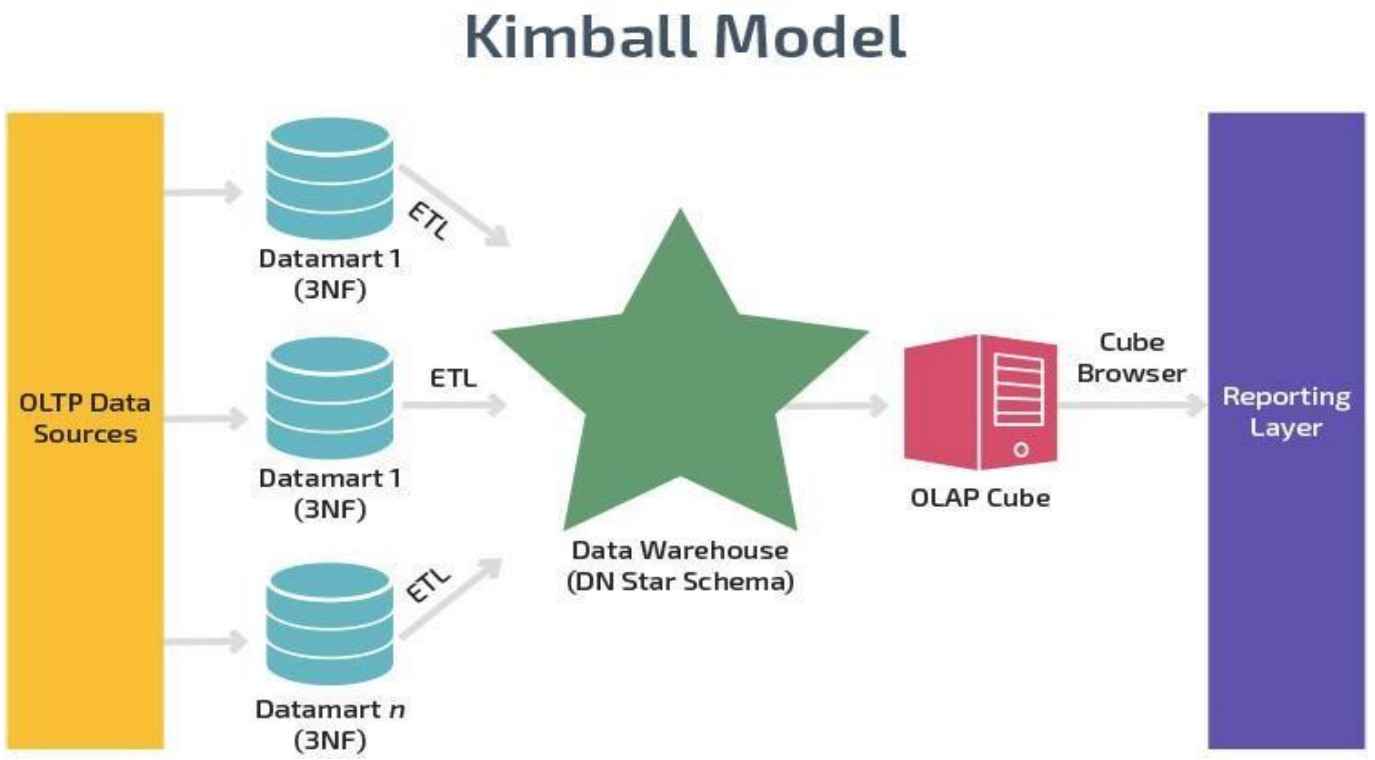
Данные жестко нормализованы даже в хранилище, витрина представляет собой физическое представление модели данных



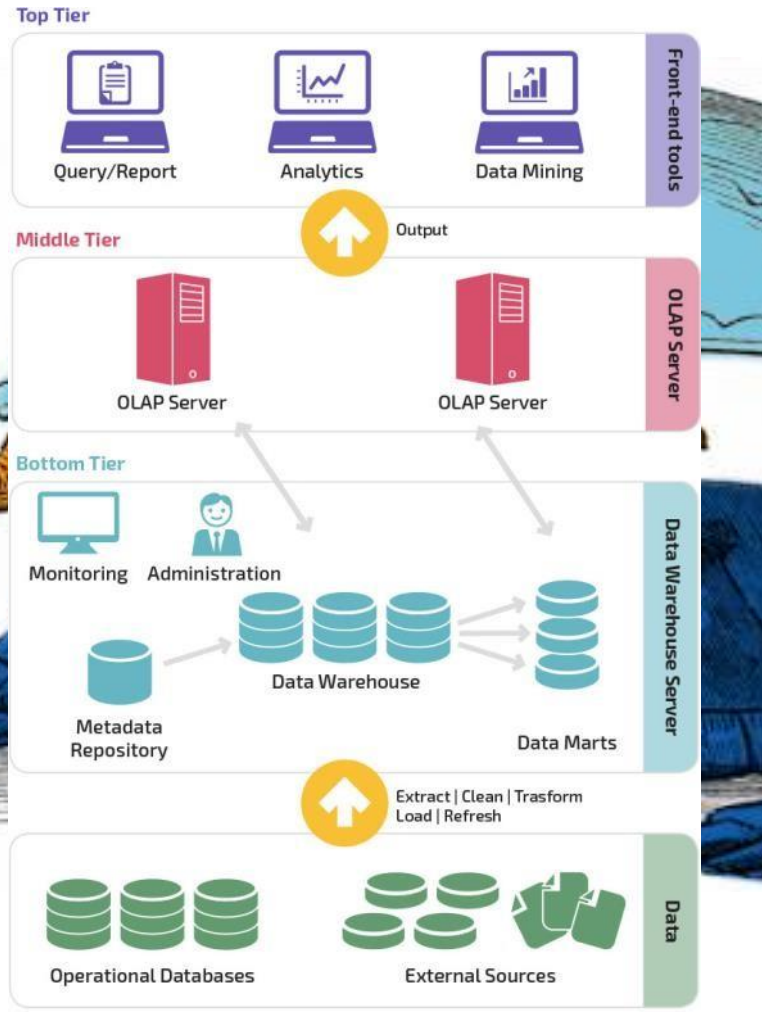
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

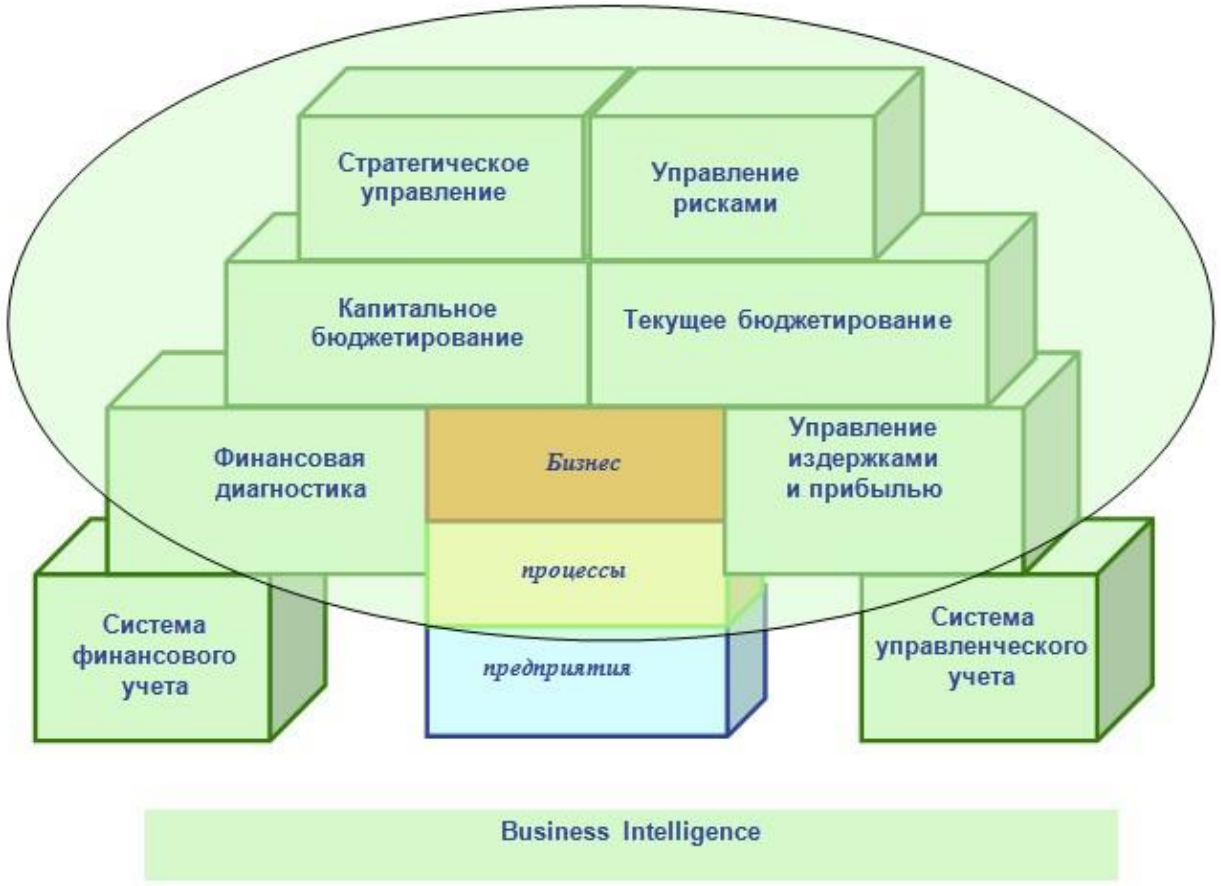
ИММОН VS КИМБАЛЛ 2/2

Данные представляют собой агрегаты и Хранилище представляет собой Конгломерат витрин данных объединённых между собой



BI (BUSINESS INTELLIGENCE)





ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

SQL (Structured Query Language)

- **Представляет собой совокупность операторов, инструкций, вычисляемых функций.**
- **Операторы SQL делятся на:**
  - операторы определения данных (Data Definition Language, DDL):
    - CREATE создаёт объект базы данных (саму базу, таблицу, представление, пользователя и так далее),
    - ALTER изменяет объект,
    - DROP удаляет объект;
  - операторы манипуляции данными (Data Manipulation Language, DML):
    - SELECT выбирает данные, удовлетворяющие заданным условиям,
    - INSERT добавляет новые данные,
    - UPDATE изменяет существующие данные,
    - DELETE удаляет данные;
  - операторы определения доступа к данным (Data Control Language, DCL):
    - GRANT предоставляет пользователю (группе) разрешения на определённые операции с объектом,
    - REVOKE отзывает ранее выданные разрешения,
    - DENY задаёт запрет, имеющий приоритет над разрешением;
  - операторы управления транзакциями (Transaction Control Language, TCL):
    - COMMIT применяет транзакцию,
    - ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции,
    - SAVEPOINT делит транзакцию на более мелкие участки.

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru