ТЕХНОЛОГИИ АНАЛИЗА БОЛЬШИХ ДАННЫХ

"Мир специально задуман так, чтобы любое познавательное усилие ума становилось новой цепью загадок«. В. Пелевин

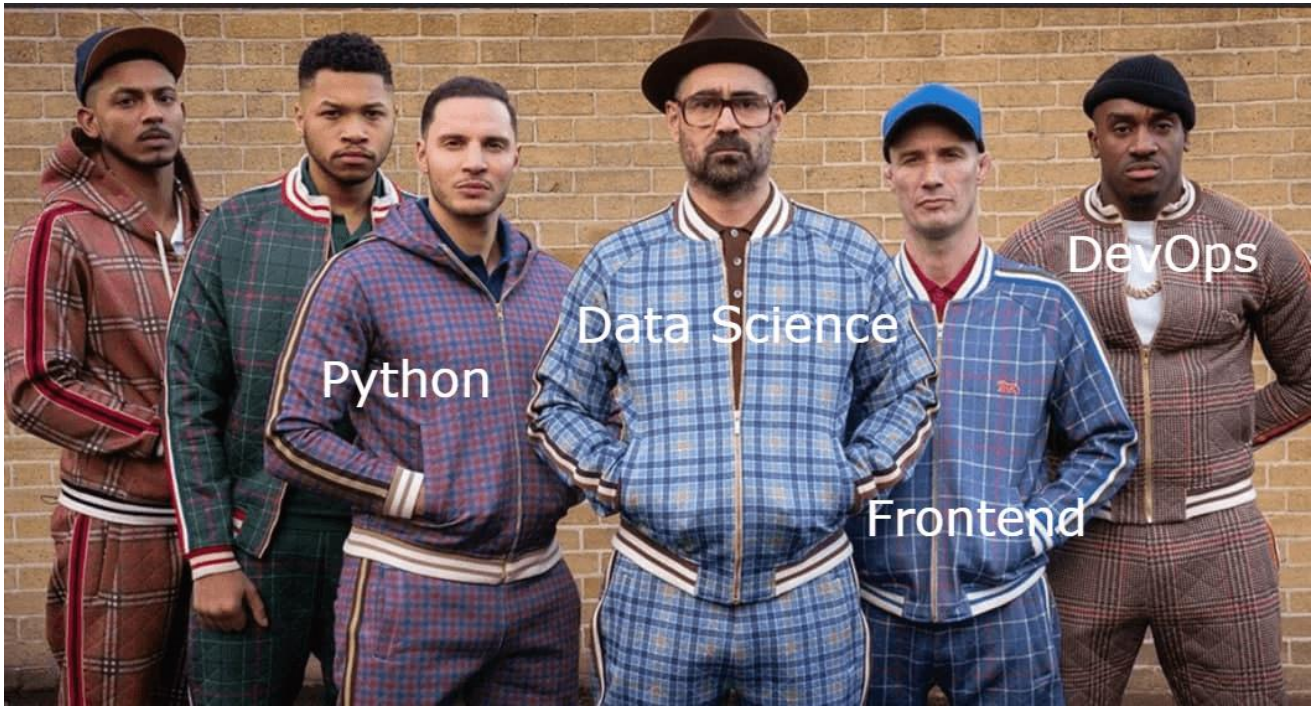
Объем человеческих знаний удваивается примерно каждые пять лет, причем время этого удвоения постоянно уменьшается. На переломе XIX–XX веков период этот составлял около пятидесяти лет. Ежедневно в мире публикуется 7 тысяч статей, печатается более 300 миллионов газет, а книг — 250 тысяч, радиоприемников и телевизоров эксплуатируется уже около 640 миллионов. Поскольку эти данные четырехлетней давности, они наверняка являются заниженными, особенно из-за стремительного роста знаний благодаря спутниковому телевидению.

Станислав Лем. **из статьи "Информационный барьер?" 1993 год.**

ГЕОРГИ МИХАИЛ ЮРЬЕВИЧ, **ИАСБ ИКТИБ**, MGEORGI@SFEDU.RU;

MOTIVATION

Data science	250 000–700 000
Аналитик	150 000–500 000
Data Engineer	200 000–400 000
BI-специалист	200 000–350 000
ETL	100 000–250 000



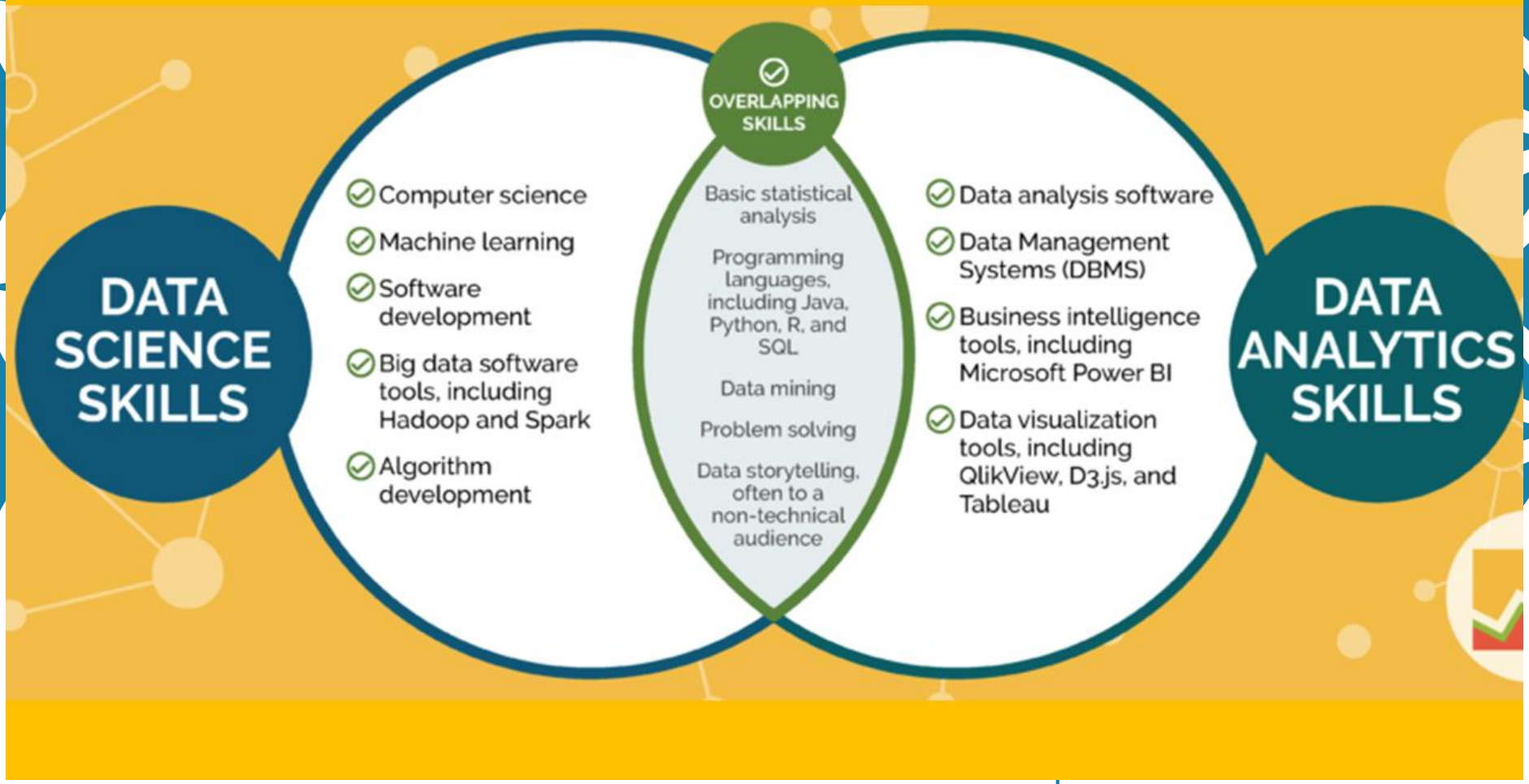
Специалисты по аналитике данных:

на 2019 год рынок специалистов с сильными компетенциями выглядел примерно так

- В 2020 на «Уроке цифры» вице-премьер Дмитрий Чернышенко заметил, что сейчас не хватает 150 000 айтишников и к 2024 году это число вырастет до 300 000.
- Минцифры: к 2027 году Россия может недосчитаться двух миллионов айтишников.

- По данным Работа.ру, средняя зарплата дата-сайентиста весной 2021 года составляла 174 000 рублей.
- Хабр.Карьера даёт результаты чуть скромнее, в районе 141000 рублей.

КТО ТАКИЕ СПЕЦИАЛИСТЫ ПО РАБОТЕ С ДАННЫМИ



КТО ТАКИЕ СПЕЦИАЛИСТЫ ПО РАБОТЕ С ДАННЫМИ

ETL-специалист — занимается выгрузкой, преобразованием и загрузкой данных из разных источников для предоставления в нужных форматах разным специалистам.

Специалист по Big Data — проектирует и создает хранилища и базы для хранения больших данных, которые можно масштабировать.

Data Engineer — выстраивает и сопровождает процессы накопления, сбора и преобразования данных. Как правило отвечает за скорость и удобство работы с данными.

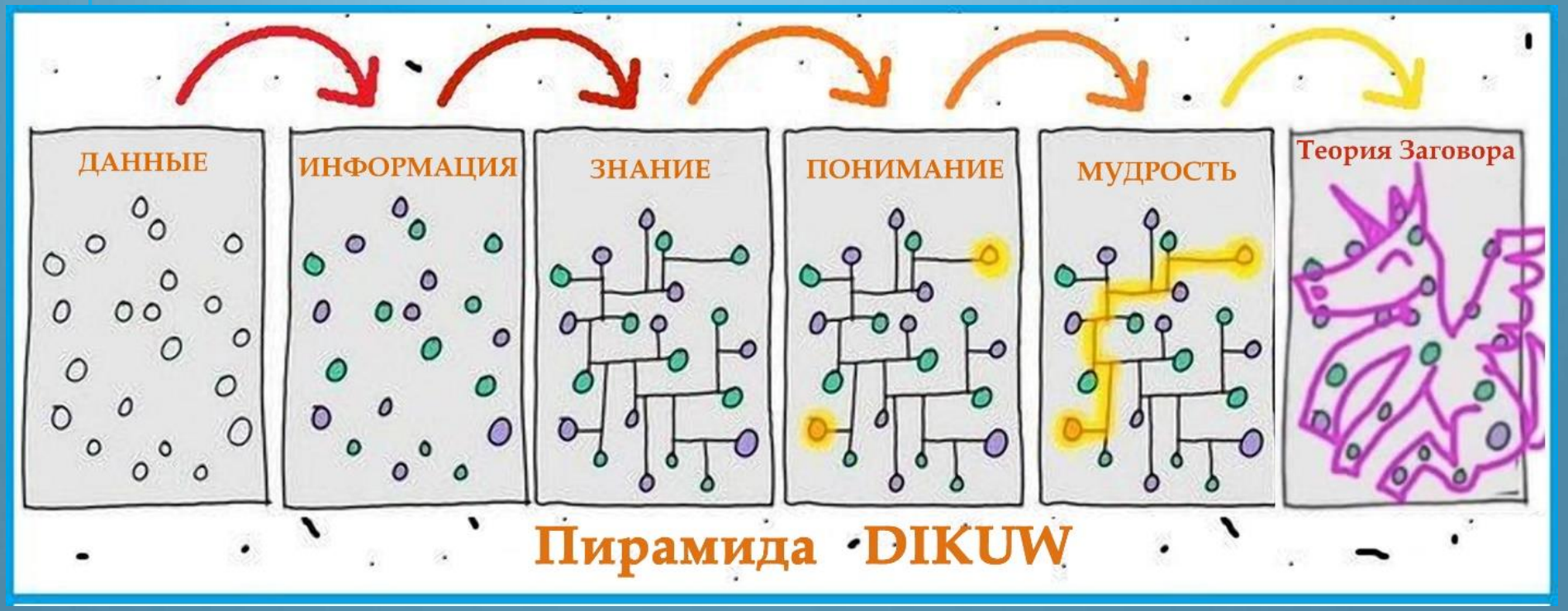
Data analyst (аналитик) — занимается непосредственно анализом данных, делает выводы на имеющихся данных, проводит АБ-тесты.

BI-analyst — отвечает за бизнес-ориентированный анализ данных и построение качественной отчетности, визуализацию.

Специалист по Machine Learning (машинному обучению) — разрабатывает и внедряет алгоритмы машинного обучения в программное обеспечение.

Data scientist — это специалист широкого профиля, он обладает экспертизой и в машинном обучении, и в аналитике, развивает data-driven продукты, позволяющие принимать решения на основе данных.

“The greatest enemy of knowledge is not ignorance; it is the illusion of knowledge.” - Stephen Hawking



КОГНИТИВНАЯ ПИРАМИДА

	Данные	Информация	Знания	Понимание	Мудрость
ОПРЕДЕЛЕНИЕ	символы, которые представляют эмпирические стимулы или восприятия	контекстуализированные данные	упорядоченная информация	ценностно-интерпретированное знание	комплексная оценка понимания
КОГНИТИВНЫЙ ПРОЦЕСС	Извлечение составных частей контекста	Соединение частей контекста	Объединение фрагментов контекста в сущность	Ассемблирование сущностей	Осознание цели
ВОПРОС		"где" и "когда"	"кто" и "что"	«как»	"почему"
ПЕРЕХОД		Понимание связей	Понимание шаблонов	Понимание причин	Понимание принципов
ПРОСТРАНСТВО	наблюдений	измерений	идей	опыта	смыслов
ПРИЧИННОСТЬ	Детерминированный процесс	Детерминированный процесс	Детерминированный процесс	интерполяционный и вероятностный процесс	интерполяционный и вероятностный процесс

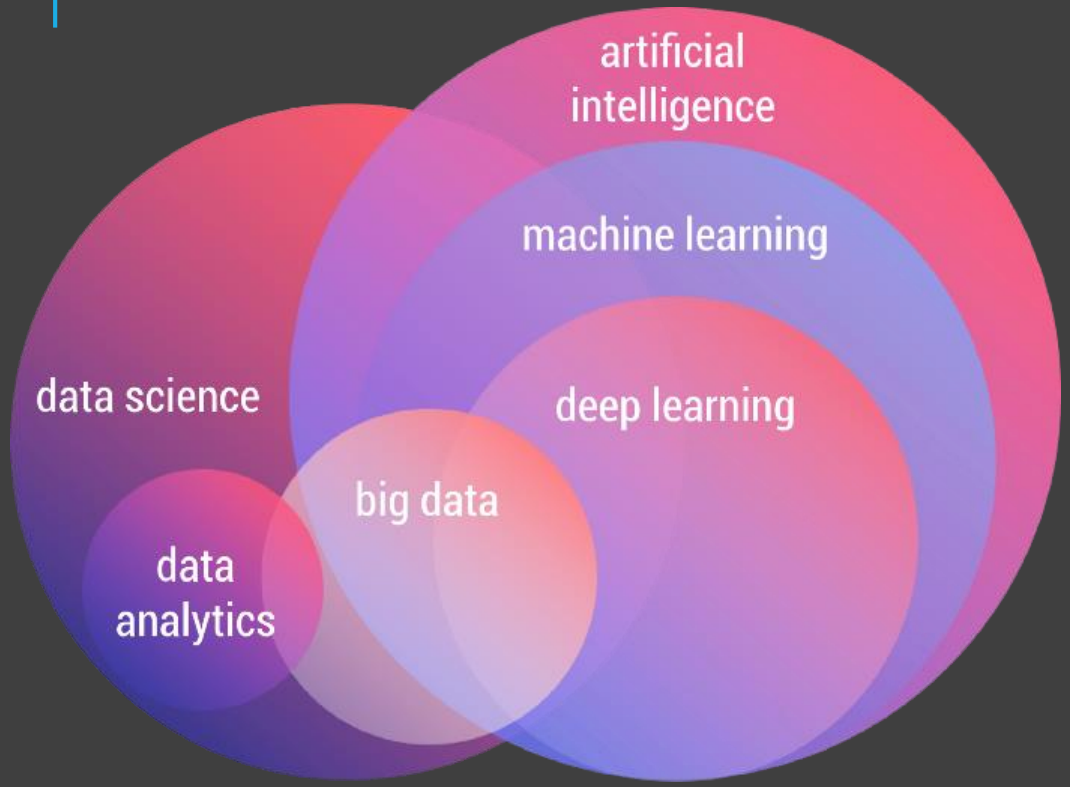
	Данные	Информация	Знания	Понимание	Мудрость
КОГНИТИВНЫЙ УРОВЕНЬ	Символьный /Явное знание	символьный/ Явное знание	субсимвольный/Явное знание	субсимвольное/Неявное знание	субсимвольное/Неявное знание
ПРЕДСТАВЛЕНИЕ	Данные представляют собой факт или утверждение о событии, не связанное с другими вещами	обработанные структурированные данные и эмпирические знания	Идентификация предметов и ситуаций. Осознание границ. Запоминание. *1	Постижение природы вещей и причинно-следственных связей. Понимание причин	Использовать свои знания и опыт для принятия правильных решений и суждений (Cambridge dictionary)
ПРИМЕР	Пример: идет дождь	Температура упала на 15 градусов, а затем пошел дождь.	Если влажность очень высока, а температура значительно падает, атмосфера часто не сможет удерживать влагу, поэтому идет дождь	Дождь идет, потому что я, невзирая на тучи, не взял зонт	Идет дождь, потому что идет дождь *2.

- Первый опыт в Data Processing датируется IV тысячелетием до нашей эры, когда появилось пиктографическое письмо. Роль данных в науке стала предметом обсуждения очень давно — первым об обработке данных еще в XVIII веке писал английский астроном Томас Симпсон в труде «О преимуществах использования чисел в астрономических наблюдениях»

- «Большие данные — это ЧТО, но не ПОЧЕМУ. И нам не всегда нужно знать причину явления; скорее, мы можем позволить данным говорить самим за себя».

- Анализ больших данных можно рассматривать как новый эпистемологический подход, смягчая требование строгой причинности в познании мира до более простой корреляции.

ТАКСОНОМИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И НАУКИ О ДАННЫХ



Дайон Хинчклиф три группы Big Data:

- Быстрые Данные (Fast Data) терабайты; не предполагает получения новых знаний, ее результаты соотносятся с априорными знаниями
- Большая Аналитика (Big Analytics) — петабайты; информации из данных преобразуется новое знание
- Глубинное Понимание (Deep Insight) — экзабайты, зеттабайты; возможно обнаружение знаний и закономерностей, априорно неизвестных.

ЦИФРОВАЯ ТРАНСФОРМАЦИЯ

Data is the new oil." — Clive Humby

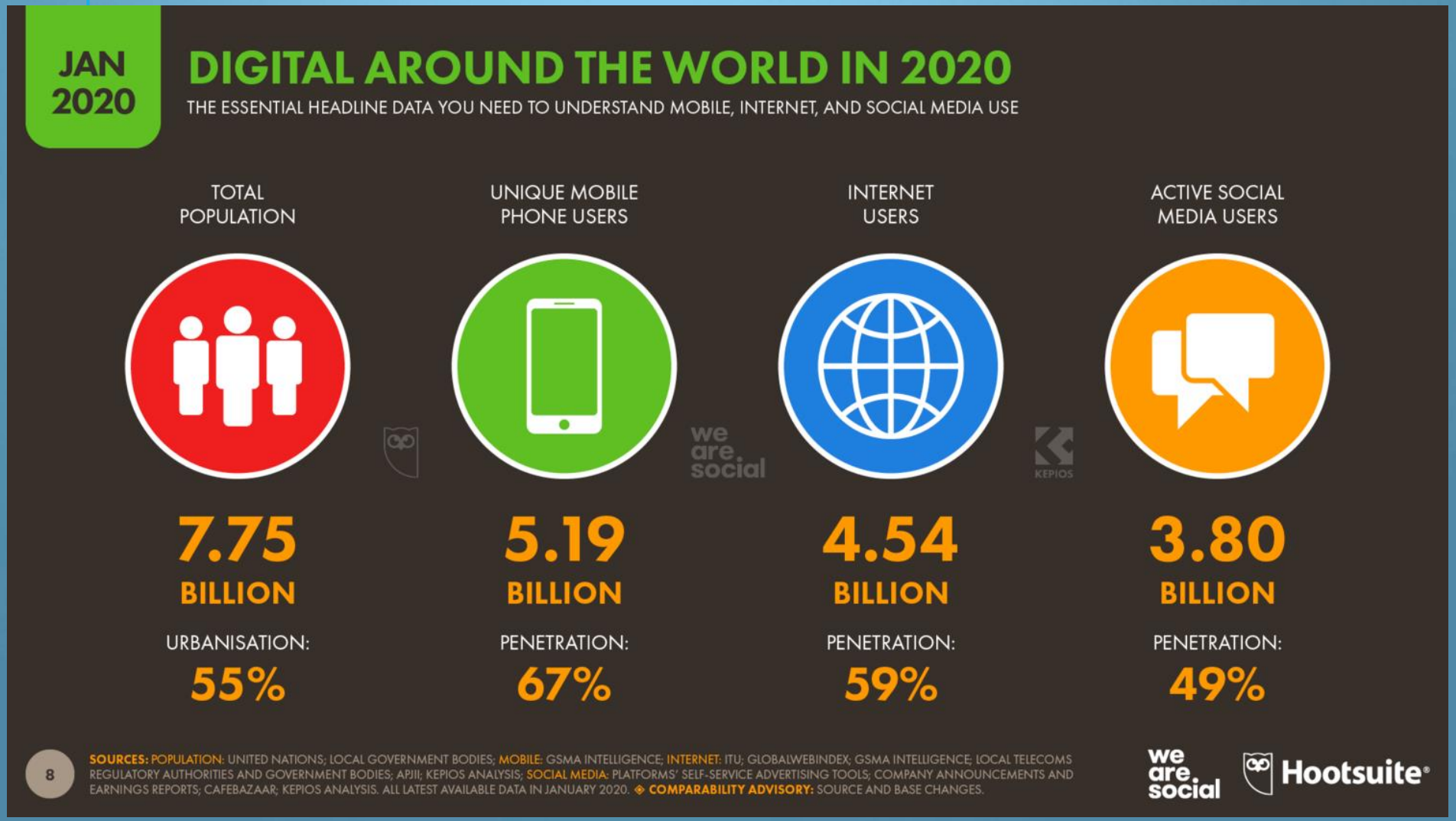
Данные — это пресловутая новая нефть и источник жизненной силы Индустрии 4.0. Как и любой другой бизнес-актив, данные следует защищать, хранить и надлежащим образом оценивать.

ТЕРМИН БОЛЬШИЕ ДАННЫЕ НЕ ИМЕЕТ ОБЩЕПРИНЯТОГО ОПРЕДЕЛЕНИЯ. НО!

Мотивация бизнеса	Пример
Желание оптимизировать бизнес-операции	Продажи, ценообразование, рентабельность, эффективность Пример: amazon.com, Walmart
Желание идентифицировать бизнес-риск	Отток клиентов, мошенничество, дефолт Пример: страхование, банковское дело
Прогнозирование новых возможностей для бизнеса	Допродажа, перекрестные продажи, поиск новых клиентов Пример: amazon.com
Соответствие законам или нормативным требованиям	Борьба с отмыванием денег, справедливое кредитование, Базель II (Операционное управление в банках) Пример: финансы

- (определение № 1) «данные очень большого размера, обычно в той степени, в которой их манипулирование и управление представляют собой значительные логистические проблемы».
- (определение №2) «всеобъемлющий термин для любой коллекции наборов данных, настолько больших и сложных, что становится трудно обрабатывать с помощью имеющихся инструментов управления данными или традиционных данных. обработка заявок »

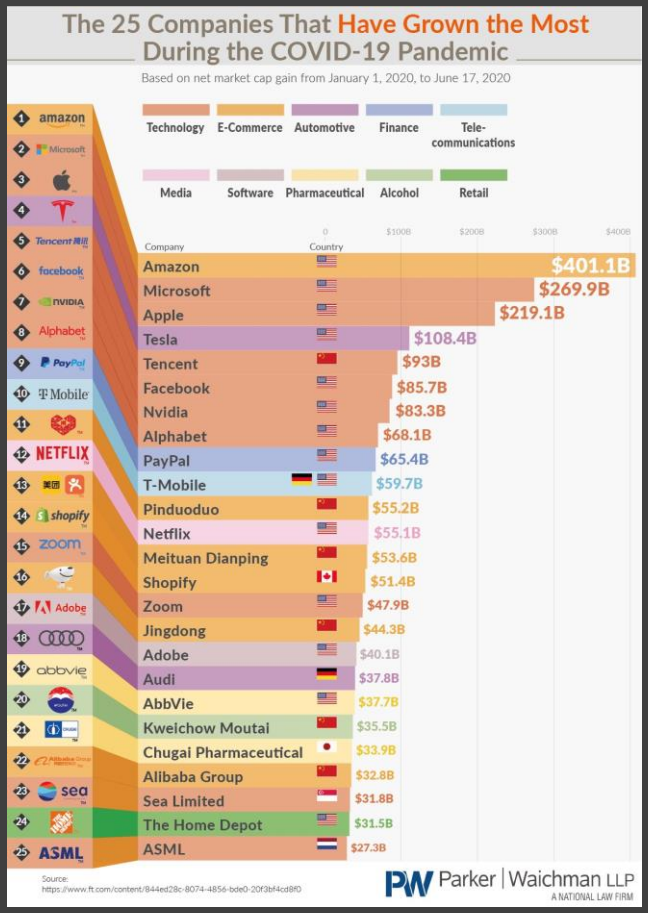
ЦИФРОВАЯ ТРАНСФОРМАЦИЯ



САМЫЕ ВАЖНЫЕ ЦИФРЫ ИЗ ГЛОБАЛЬНОГО ОТЧЕТА DIGITAL 2020

ЦИФРОВАЯ ТРАНСФОРМАЦИЯ

- **Обработка больших объемов данных — основа цифровой трансформации, и ключом к ее реализации является концепция озер данных, хранилищ данных, а также хабов и витрин данных.**



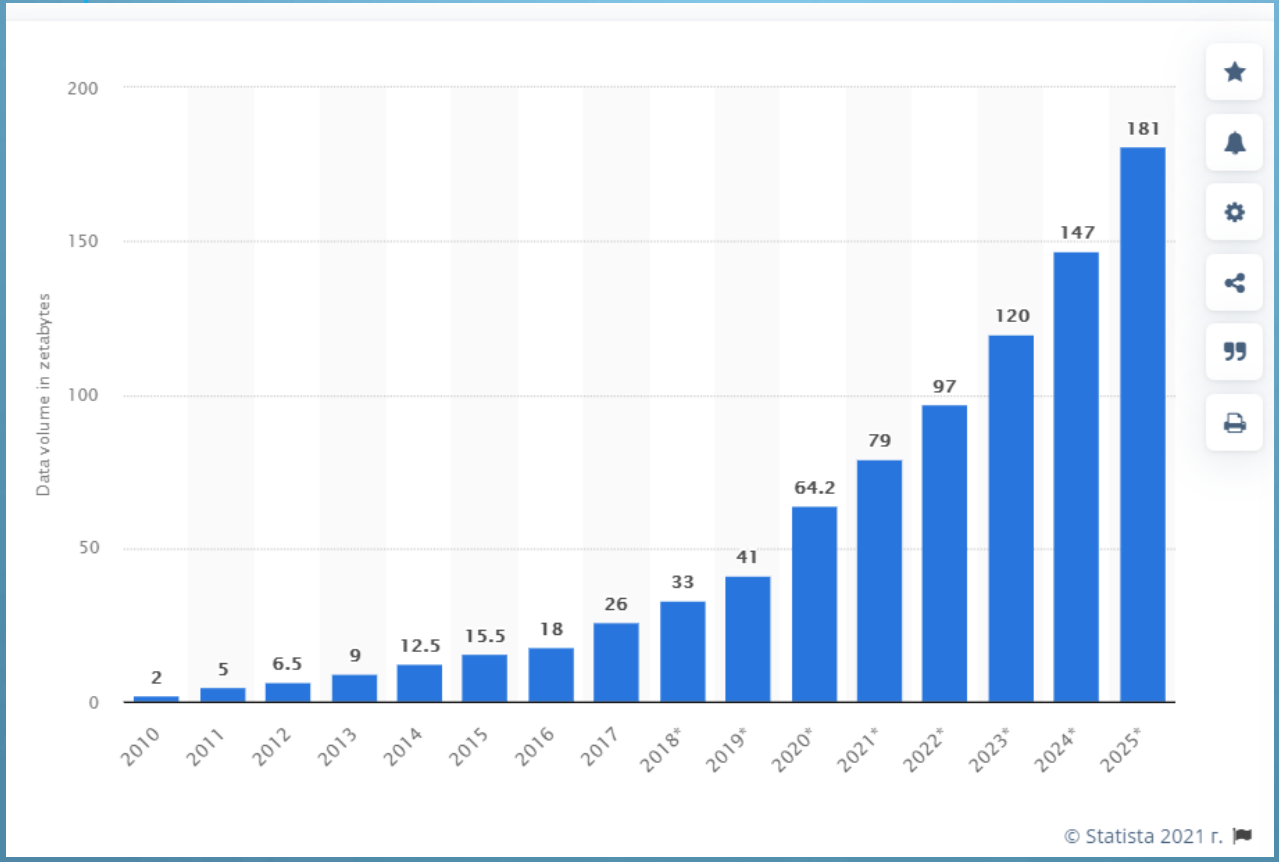
Визуальная иллюстрация количества данных, создаваемых за минуту (согласно Domo, апрель 2019 г.)

Источники порождающие Большие Данные:

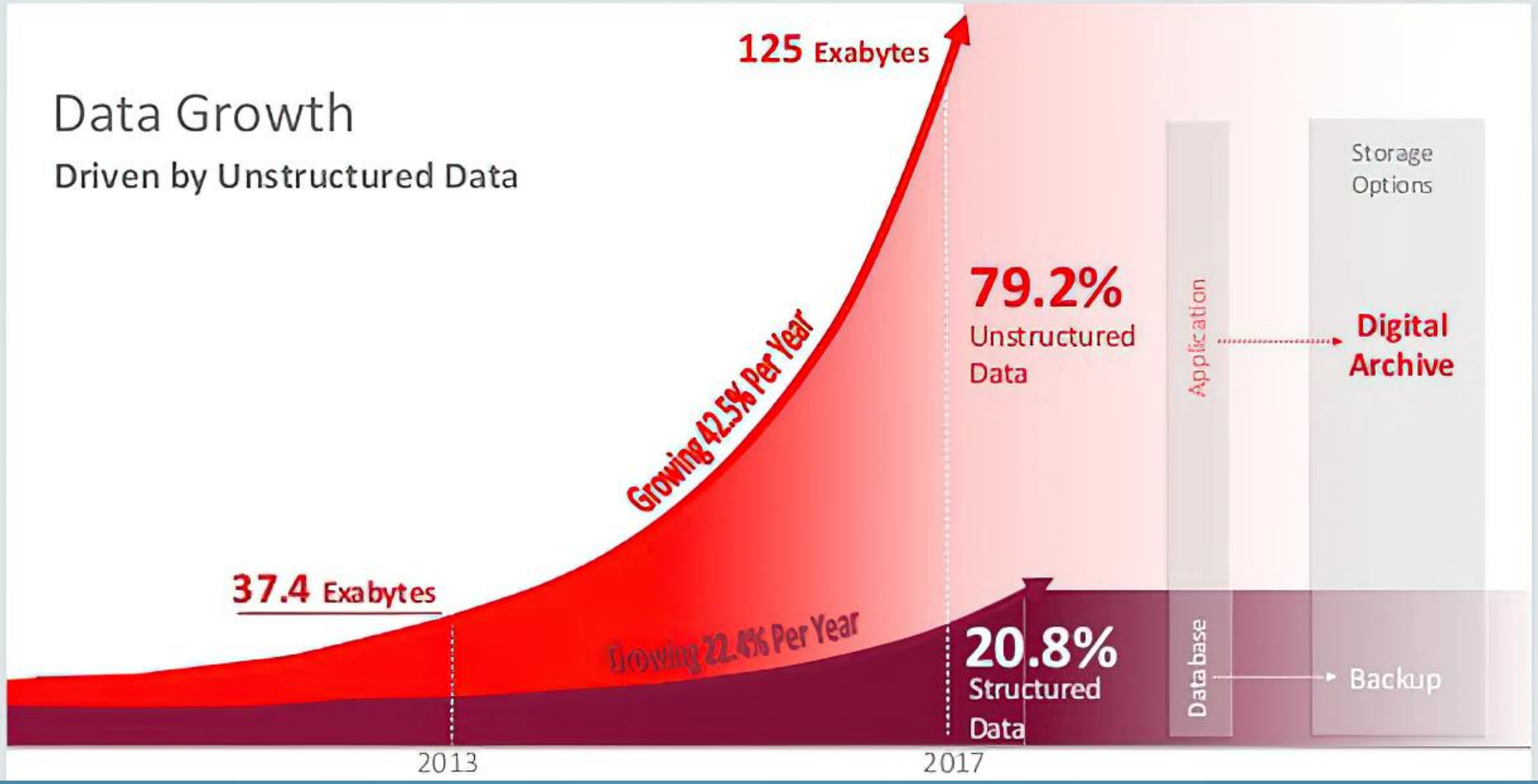
- **социальные сети — посты, комментарии, сообщения между пользователями и пр.;**
- **события, связанные с действиями пользователей в веб- или мобильных приложениях;**
- **логи приложений;**
- **телеметрия сети устройств из мира «Интернета вещей» (Internet of Things, IoT);**
- **потоки событий крупных веб-приложений;**
- **потоки транзакций банковских платежей с метаданными (время, место платежа и т. д.).**
- **Большой адронный коллайдер, Youtube**

ДАННЫЕ С ТОЧКИ ЗРЕНИЯ БИЗНЕСА

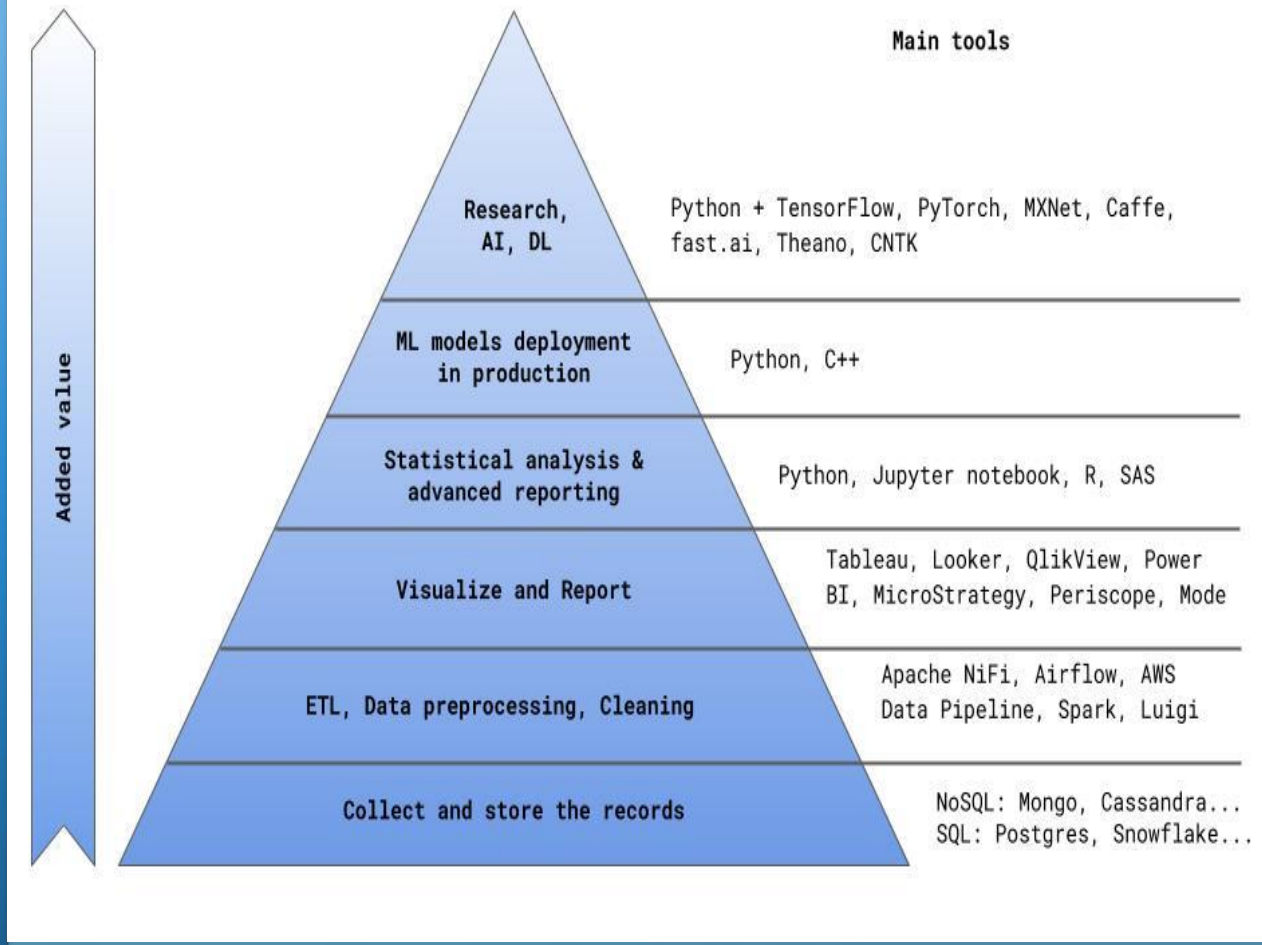
Data is the new oil.” — Clive Humby



Объем данных , созданных, собранных, копированных и потребленных во всем мире с 2010 по 2025 год в зеттабайтах


Data Growth
Driven by Unstructured Data
125 Exabytes
37.4 Exabytes
Growing 42.5% Per Year
Growing 22.4% Per Year
79.2%
Unstructured
Data
20.8%
Structured
Data
Application
Database
Storage
Options
Digital
Archive
Backup
2013
2017

BI система — это термин, объединяющий программные продукты, инструменты, инфраструктуру и лучшие практики, который позволяет улучшать и оптимизировать принимаемые решения

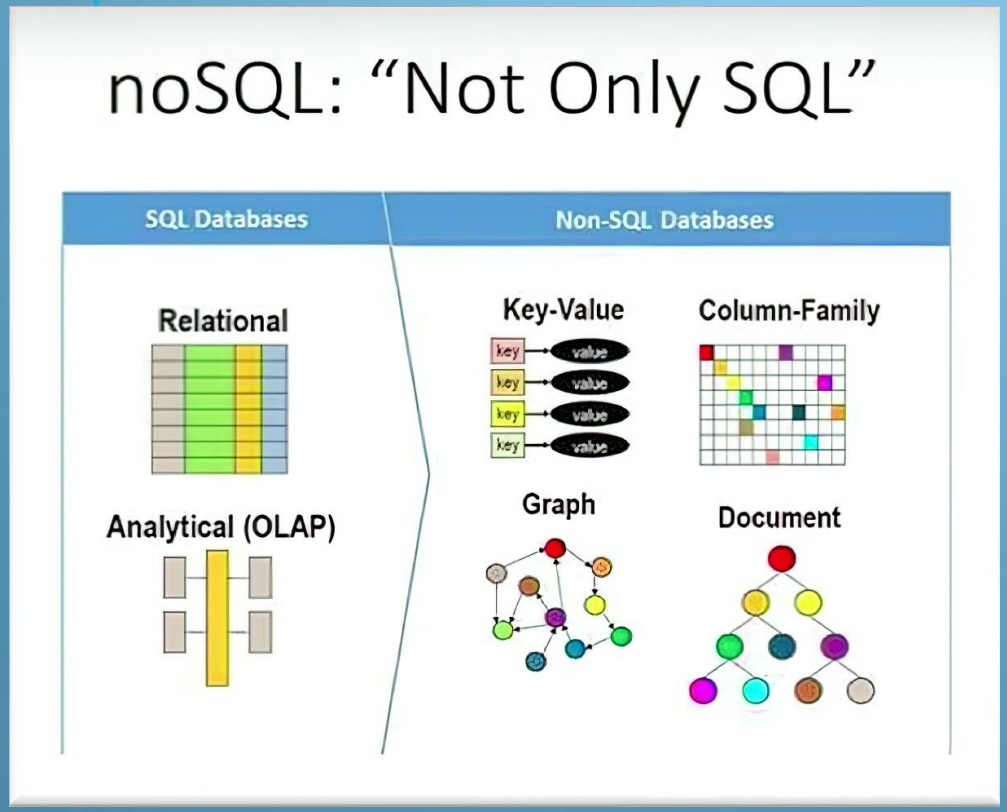


РЕЙТИНГ СУБД

380 systems in ranking, October 2021

Rank Oct 2021	Rank Sep 2021	Rank Oct 2020	DBMS	Database Model	Score Oct 2021	Score Sep 2021	Score Oct 2020
1.	1.	1.	Oracle	Relational, Multi-model	1270.35	-1.19	-98.42
2.	2.	2.	MySQL	Relational, Multi-model	1219.77	+7.24	-36.61
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	970.61	-0.24	-72.51
4.	4.	4.	PostgreSQL	Relational, Multi-model	586.97	+9.47	+44.57
5.	5.	5.	MongoDB	Document, Multi-model	493.55	-2.95	+45.53
6.	6.	↑ 8.	Redis	Key-value, Multi-model	171.35	-0.59	+18.07
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	165.96	-0.60	+4.06
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model	158.25	-1.98	+4.41
9.	9.	9.	SQLite	Relational	129.37	+0.72	+3.95
10.	10.	10.	Cassandra	Wide column	119.28	+0.29	+0.18
11.	11.	11.	Microsoft Access	Relational	116.38	-0.56	-1.87

ТИПЫ СУБД



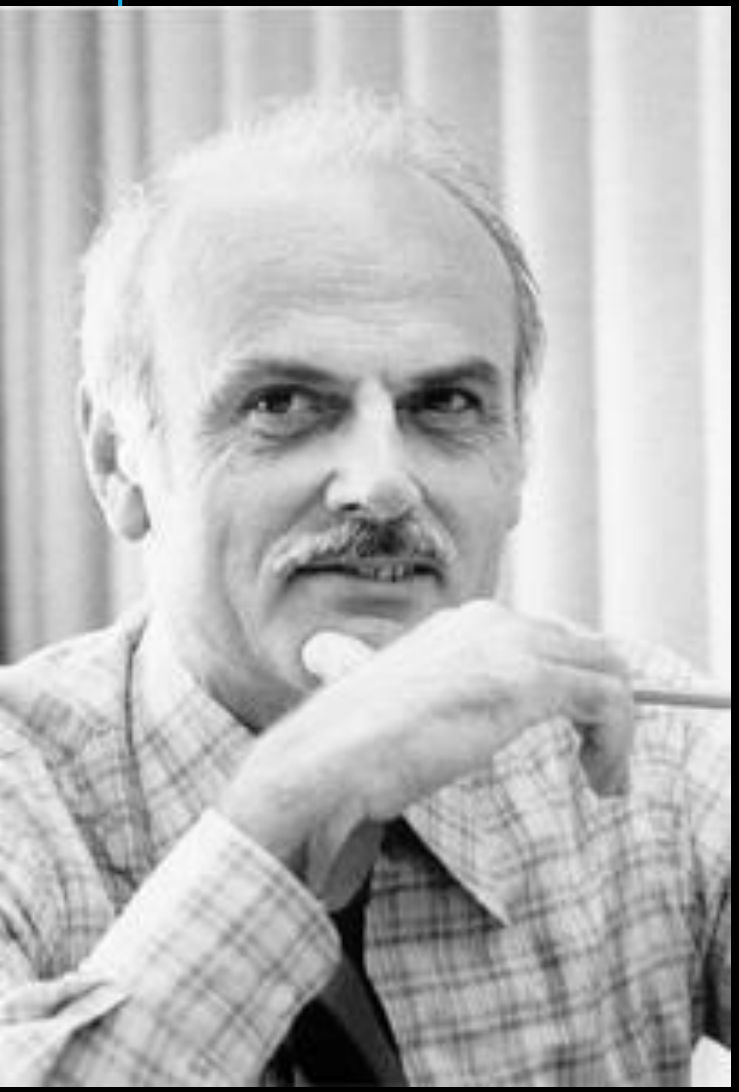
1. MongoDB - самая популярная база данных NoSQL на основе документов.
2. ElasticSearch - Эта база данных NoSQL используется, если полнотекстовый поиск является частью вашего решения.
3. DynamoDB - База данных Amazon NoSQL известна своей масштабируемостью..
4. Hbase - Это хорошо масштабируемая распределенная система баз данных с открытым исходным кодом
5. Cassandra - Это решение для базы данных было впервые создано Facebook.

Relational DBMS

Первые системы управления реляционными базами данных появились на рынке в начале 1980-х годов и с тех пор являются наиболее часто используемым типом СУБД

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM Db2

- Системы управления реляционными базами данных (СУБД) поддерживают реляционную (= таблично-ориентированную) модель данных.
- Схема таблицы (= схема отношения) определяется именем таблицы и фиксированным количеством атрибутов с фиксированными типами данных.
- Запись (= объект) соответствует строке в таблице и состоит из значений каждого атрибута.
- Таким образом, отношение состоит из набора однородных записей.



Реляционные базы данных. Нормализация БД

- Большинство современных систем управления базами данных (СУБД) разработаны на основе реляционной алгебры.
- Первая работа по реляционной модели данных «A Relational Model of Data for Large Shared Data Banks» была опубликована в 1970 г.
- Её автор - Эдгар Франк Кодд. В своей статье Э. Кодд вывел несколько правил, или форм, по упорядочиванию данных и их отношений.
- **Нормализация БД** - это проектирование базы данных так, чтобы она была компактной и не несла логическую избыточность. Существует несколько разновидностей нормализации, так называемые **нормальные формы**. Все они идут в порядке усложнения от простого.
- Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.
- Всего существует 6 нормальных форм. На практике редко нормализуют выше 3-ей нормальной формы.

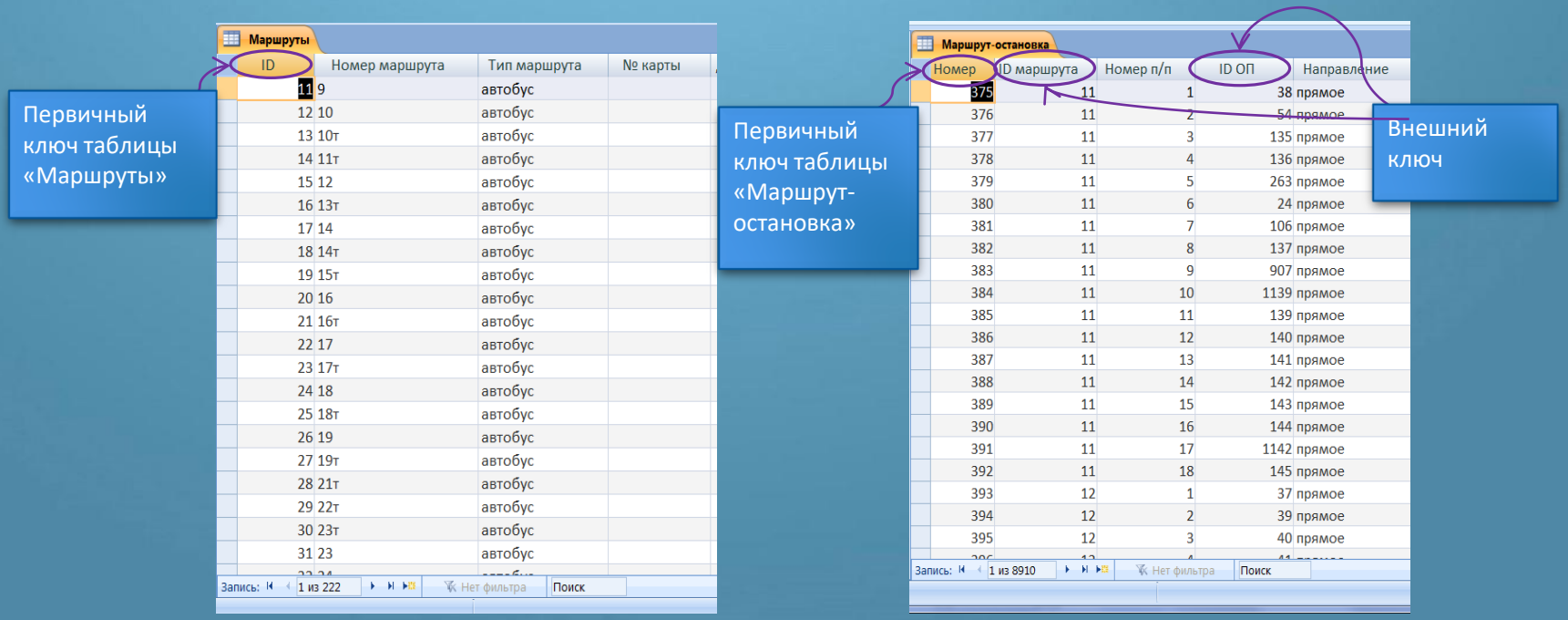
Ключи

Ключи являются составляющей частью нормализованных таблиц. Бывают двух видов — внешние и первичные.

Первичный ключ — это атрибут, значения которого уникально идентифицируют каждую запись таблицы. Первичный ключ отвечает следующим условиям:

- **Он должен иметь значение, не NULL.**
- **Быть неизменным — значение ключа не должно меняться.**
- **Иметь уникальное значение для каждой строки.**

Внешние ключи — это ссылки на первичные ключи других таблиц.



Key-value Stores

Эти простые системы обычно не подходят для сложных приложений. Их простота востребована Например, ресурсоэффективные хранилища ключей и значений часто применяются во встроенных системах или как высокопроизводительные внутрипроцессные базы данных.

- **Redis**
- **Amazon DynamoDB**
- **Microsoft Azure Cosmos DB**
- **Memcached**
- **Etcd**
- **Riak**

- **Хранилища значений ключей позволяют разработчику хранить данные без схемы.**
- **В хранилище ключ-значение база данных хранит данные в виде хеш-таблицы, где каждый ключ уникален, а значение может быть строкой, JSON, BLOB (базовый большой объект) и т. Д.**
- **Ключом могут быть строки, хэши, списки, наборы, отсортированные наборы и значения, хранящиеся по этим ключам.**
- **Например, пара "ключ-значение" может состоять из такого ключа, как "Имя", который связан со значением, например, "Робин".**
- **Хранилища ключей и значений могут использоваться как коллекции, словари, ассоциативные массивы и т. Д.**
- **Хранилища ключей и значений хорошо подходят для содержимого корзины покупок или отдельных значений, таких как цветовые схемы, URI целевой страницы или номер учетной записи по умолчанию.**

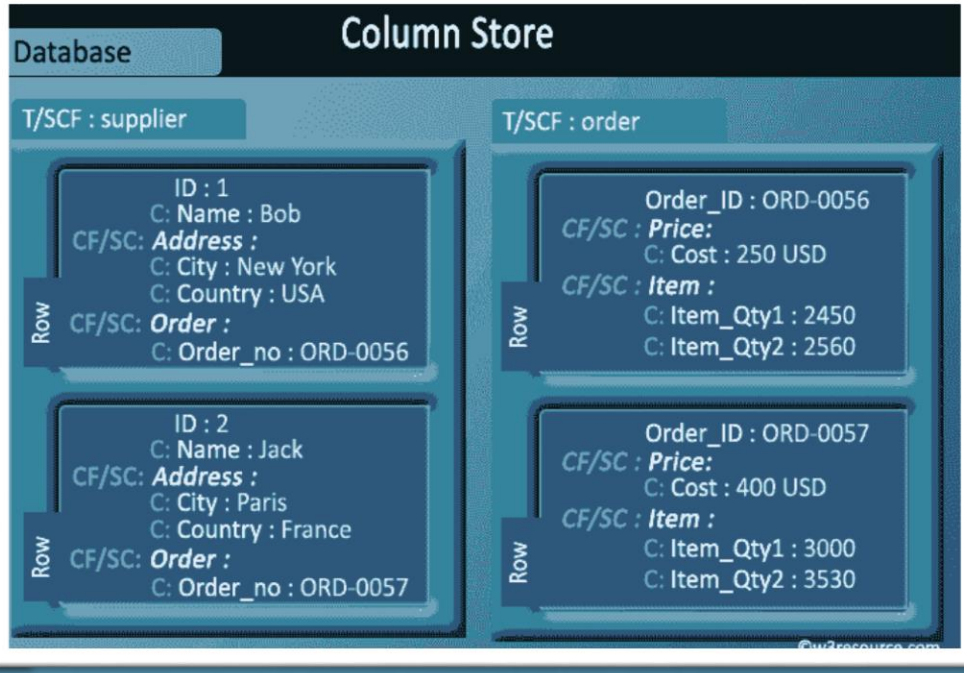
Wide Column Stores

Хранилища с широкими столбцами, также называемые расширяемыми хранилищами записей, хранят данные в записях с возможностью хранения очень большого количества динамических столбцов. Поскольку имена столбцов, а также ключи записи не фиксированы, и поскольку запись может содержать миллиарды столбцов, широкие хранилища столбцов можно рассматривать как двумерные хранилища ключей и значений .

- **Cassandra**
- **Hbase**
- **Microsoft Azure Cosmos DB**
- **Google BigTable**

- Базы данных, ориентированные на столбцы, в основном работают по столбцам, где каждый столбец обрабатывается индивидуально.
- Значения одного столбца хранятся непрерывно.
- Столбец хранит данные в файлах, специфичных для столбца.
- В хранилищах столбцов обработчики запросов также работают со столбцами.
- Все данные в каждом файле данных столбца имеют один и тот же тип, что делает его идеальным для сжатия.
- Хранилища столбцов могут повысить производительность запросов, поскольку они могут получить доступ к определенным данным столбца.
- **Высокая производительность при запросах агрегирования (например, COUNT, SUM, AVG, MIN, MAX).**
- **Работает над хранилищами данных и бизнес-аналитикой, управлением взаимоотношениями с клиентами (CRM), каталогами библиотечных карточек и т. Д**

Key-value vs Column Stores



Document Stores

Эти простые системы обычно не подходят для сложных приложений. Их простота востребована Например, ресурсоэффективные хранилища ключей и значений часто применяются во встроенных системах или как высокопроизводительные внутрипроцессные базы данных.

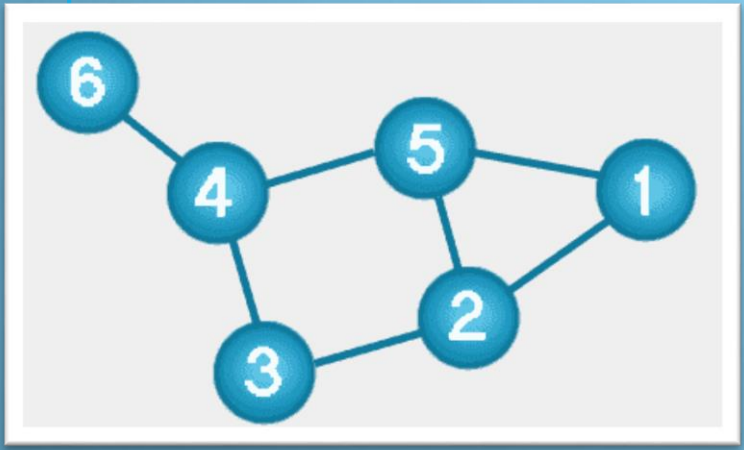
- MongoDB
- Amazon DynamoDB
- Microsoft Azure Cosmos DB
- Couchbase
- Firebase Realtime Database

- Коллекция документов
- Данные в этой модели хранятся внутри документов.
- Документ — это набор значений ключа, в котором ключ позволяет получить доступ к его значению.
- Документы обычно не требуют наличия схемы, поэтому их можно легко изменить.
- Документы хранятся в коллекциях, для группировки различных типов данных.
- Документы могут содержать много разных пар ключ-значение, пар ключ-массив или даже вложенных документов.

Key-value vs Column Stores

Реляционная модель	Документная модель
Tables	Collections
Rows	Documents
Columns	Key/value pairs
Joins	not available

Graph-based Stores

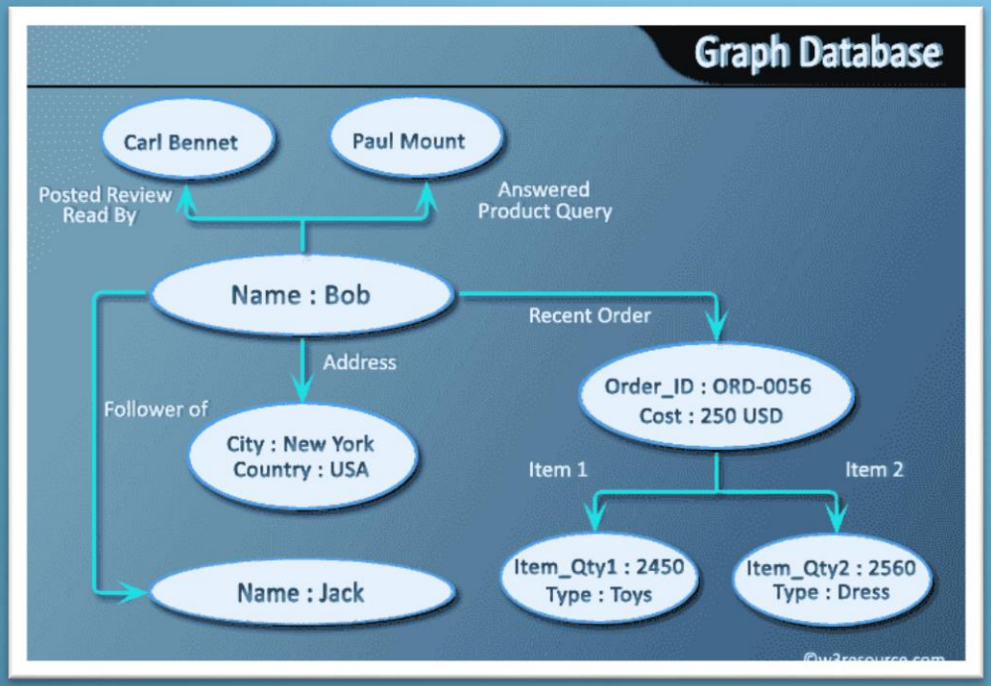


Структура данных графа состоит из конечного (и, возможно, изменяемого) набора упорядоченных пар, называемых ребрами или дугами, определенных объектов, называемых узлами или вершинами.

- **Neo4j**
- **ArangoDB**
- **OrientDB**
- **Amazon Neptune**

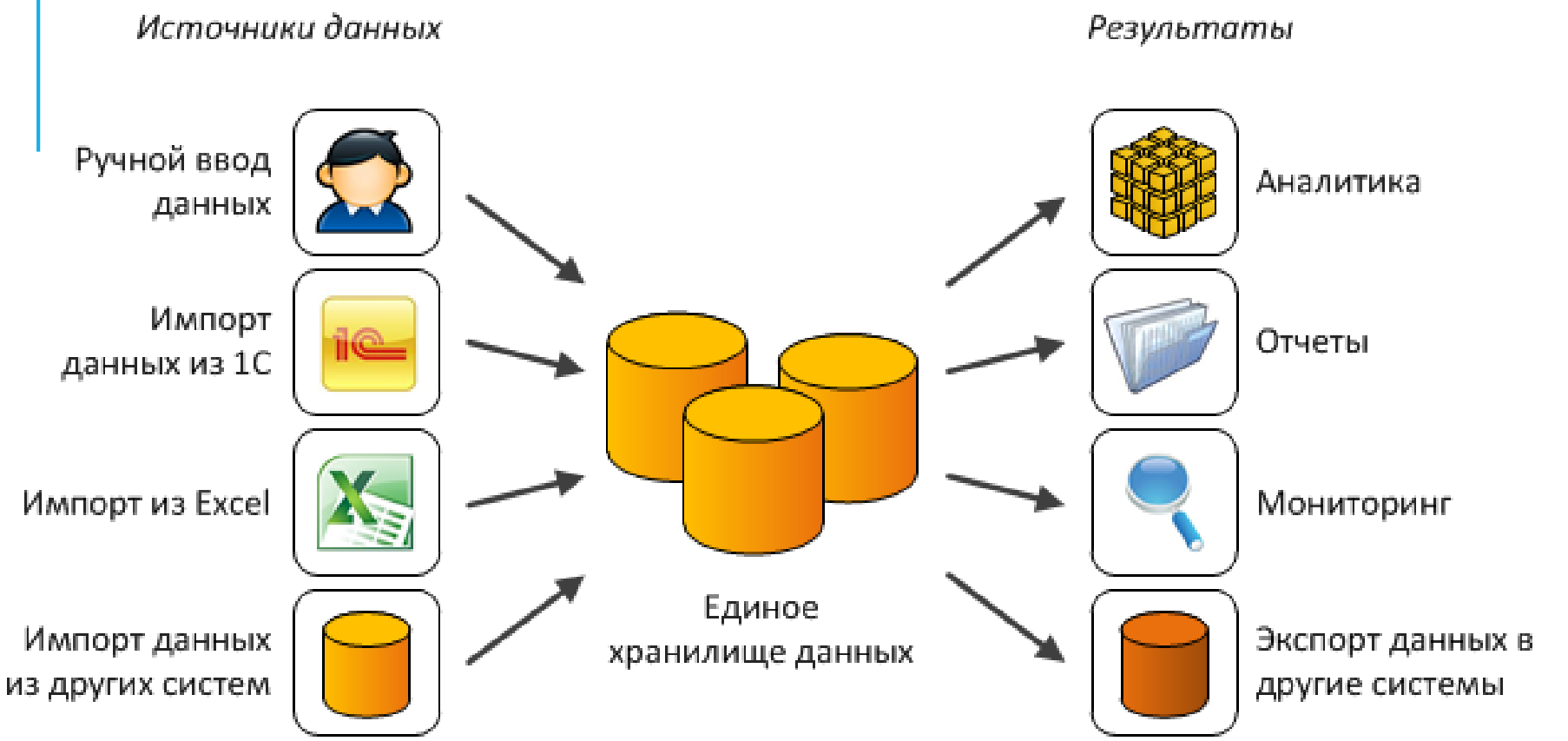
- **Хранит данные в виде графа, набор узлов и ребер**
- **Способно элегантно представлять любые данные в очень доступной форме.**
- **Каждый узел представляет собой объект (например, студента или компанию), а каждое ребро представляет собой соединение или связь между двумя узлами.**
- **Каждый узел и ребро определяется уникальным идентификатором.**
- **Каждый узел знает свои соседние узлы.**
- **По мере увеличения количества узлов стоимость локального шага (или перехода) остается неизменной.**
- **Глобальная индексация для поиска.**

Graph-based Stores



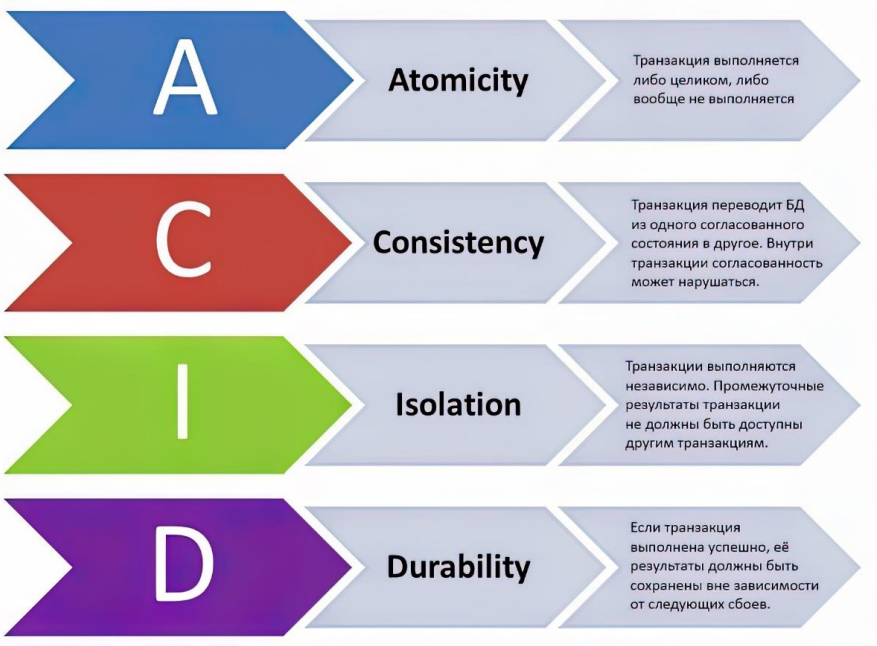
Relational model	Graph model
Tables	Vertices and Edges set
Rows	Vertices
Columns	Key/value pairs
Joins	Edges

АРХИТЕКТУРА БОЛЬШИХ ДАННЫХ. МУЛЬТИМОДАЛЬНОСТЬ



Часто в компаниях существует несколько информационных систем – системы складского учета, бухгалтерские системы, ERP системы для автоматизации отдельных производственных процессов, системы сбора отчетности с подразделений компании, а также множество файлов, которые разбросаны по компьютерам сотрудников.

OLTP И OLAP-ТЕХНОЛОГИИ.
MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP), HOLAP (Hybrid OLAP)



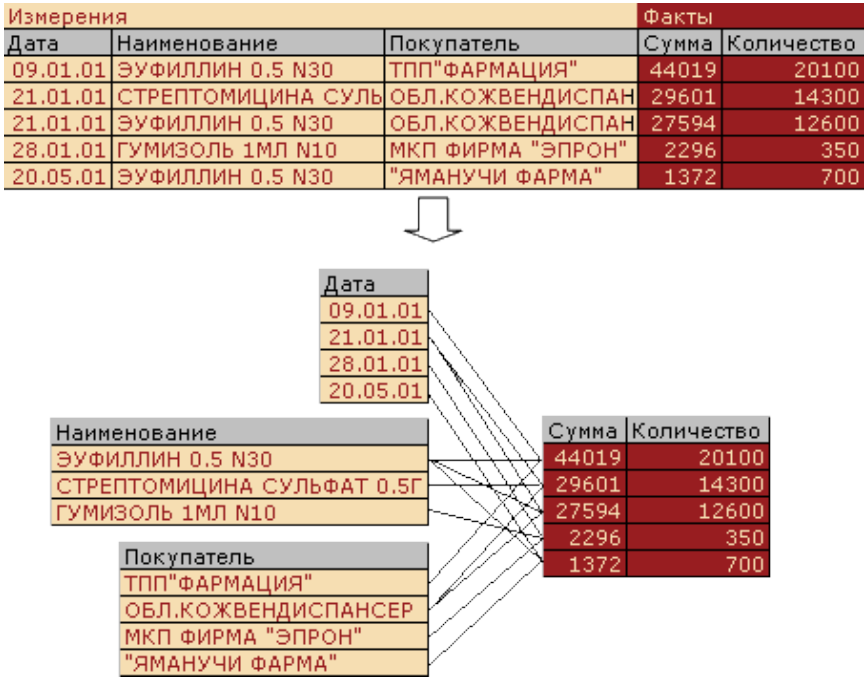
Онлайн-обработка транзакций (OLTP) собирает, хранит и обрабатывает данные транзакций в режиме реального времени. В OLTP упор делается на быструю обработку, поскольку базы данных OLTP часто читаются, записываются и обновляются. В случае сбоя транзакции встроенная системная логика обеспечивает целостность данных.

Имея столько разрозненных источников информации, часто бывает очень сложно получить ответы на ключевые вопросы деятельности компании и увидеть общую картину. А когда нужная информация все же находится в одной из используемых систем или локальном файле, то она часто оказывается устаревшей или противоречит информации, полученной из другой системы.

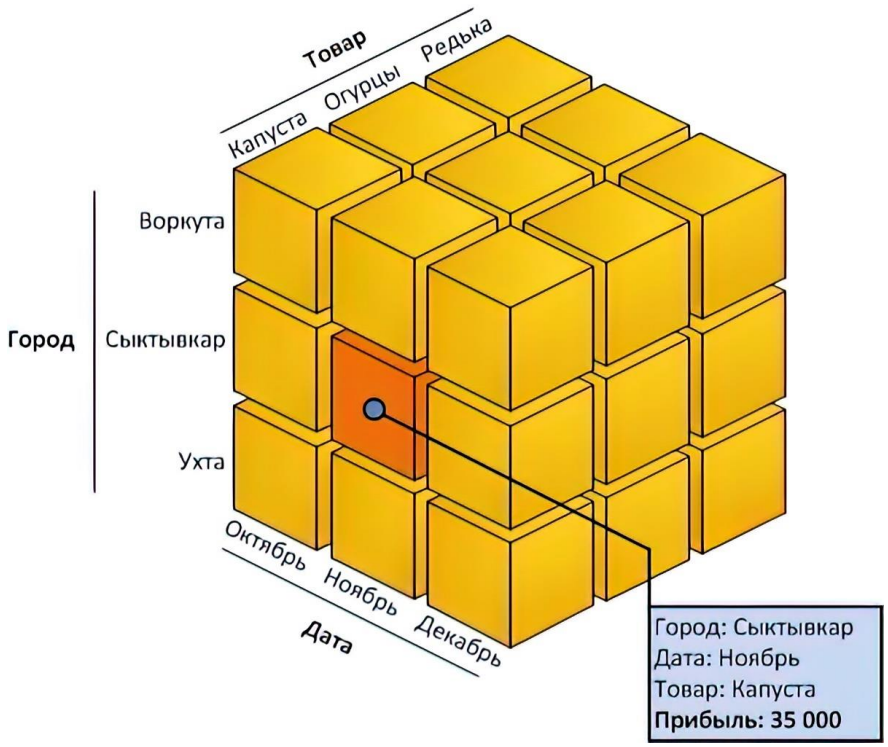
OLTP и OLAP-технологии. MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP), HOLAP (Hybrid OLAP)

OLAP (on-line analytical processing) — набор технологий для оперативной обработки информации, включающих динамическое построение отчётов в различных разрезах, анализ данных, мониторинг и прогнозирование ключевых показателей бизнеса. В основе OLAP-технологий лежит представление информации в виде OLAP-кубов. Онлайн-аналитическая обработка (OLAP) использует сложные запросы для анализа агрегированных исторических данных из OLTP-систем.

OLAP-системы, самой идеологией своего построения предназначены для анализа больших объёмов информации, позволяют преодолеть ограничения традиционных информационных систем.



OLTP И OLAP-ТЕХНОЛОГИИ.
MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP), HOLAP (Hybrid OLAP)



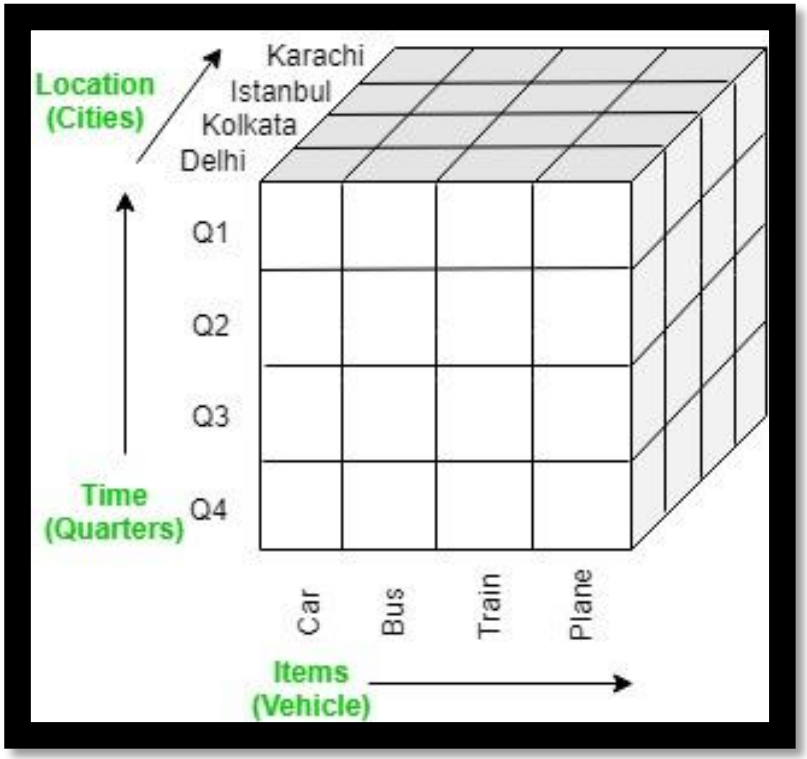
- Интегрировать данные различных информационных систем, создав единую версию правды
- Проектировать новые отчеты несколькими щелчками мыши без участия программистов.
- В реальном времени анализировать данные по любым категориям и показателям бизнеса на любом уровне детализации.
- Производить мониторинг и прогнозирование ключевых показателей бизнеса.

Бизнес-показатели хранятся в кубах не в виде простых таблиц, как в обычных системах учета или бухгалтерских программах, а в разрезах, представляющих собой основные бизнес-категории деятельности организации: товары, магазины, клиенты, время продаж и т. д.

Операции OLAP.

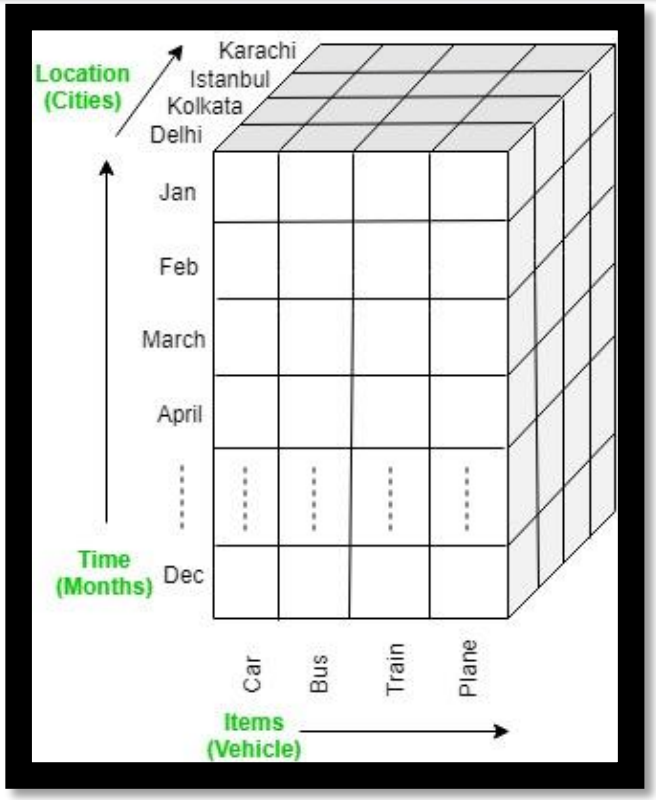
Есть пять основных аналитических операций, которые можно выполнить с кубом OLAP:

- Базы данных OLAP разделены на один или несколько кубов, и эти кубы известны как гиперкубы.
- Позволяет пользователю запрашивать многомерные данные (например, Дели -> 2018 -> Данные о продажах)



Детализация (Drill down): в операции детализации менее подробные данные преобразуются в высокодетализированные. Это можно сделать:

- Спуск по иерархии концепций
- Добавление нового измерения

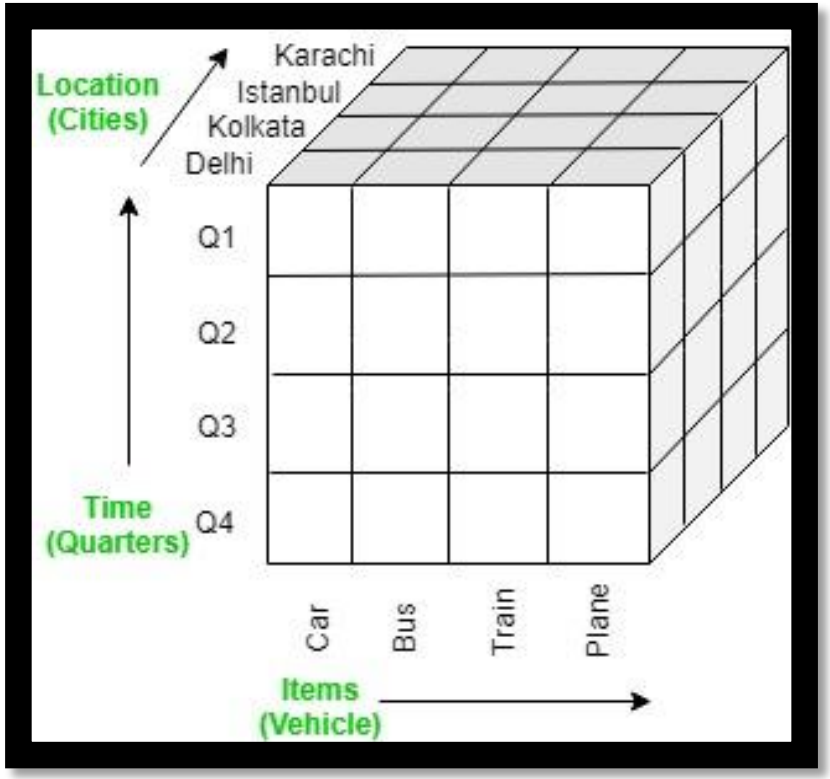


Операции OLAP.

Есть пять основных аналитических операций, которые можно выполнить с кубом OLAP:

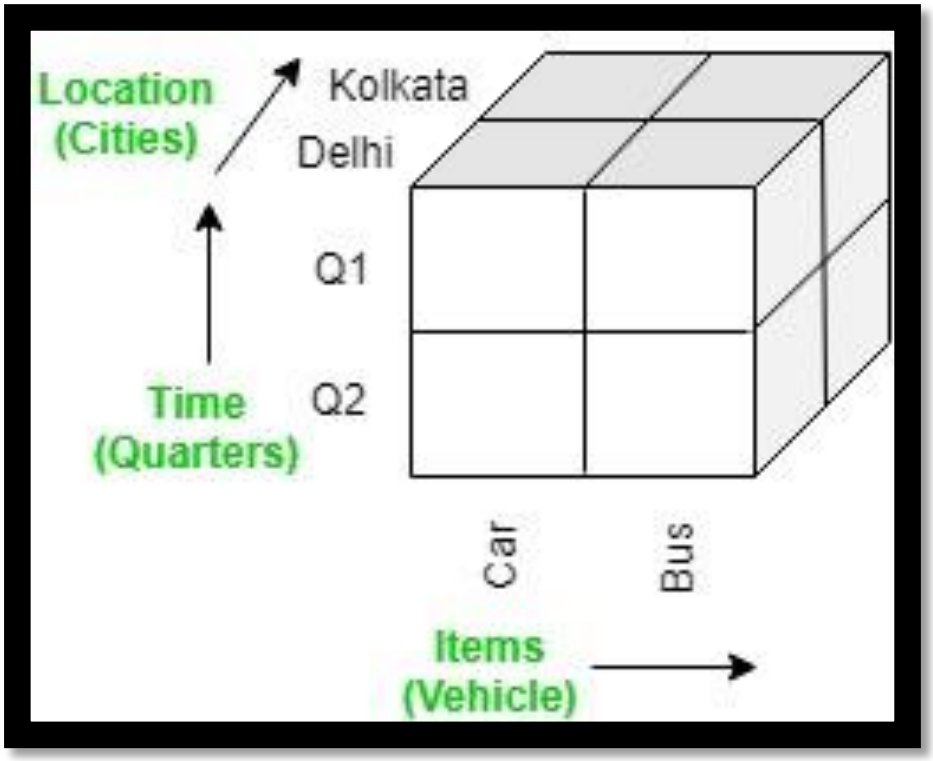
Сворачивание (Roll up): это прямо противоположно операции детализации. Он выполняет агрегирование куба OLAP. Это можно сделать:

- **Восхождение в иерархии концепций**
- **Уменьшение размерности**



Вырезы (Dice): выбирает вложенный куб из куба OLAP, выбирая два или более измерений. В кубе, приведенном в разделе обзора, субкуб выбирается путем выбора следующих измерений с критериями:

- **Местоположение = «Дели» или «Калькутта».**
- **Время = «Q1» или «Q2»**
- **Item = «Автомобиль» или «Автобус»**

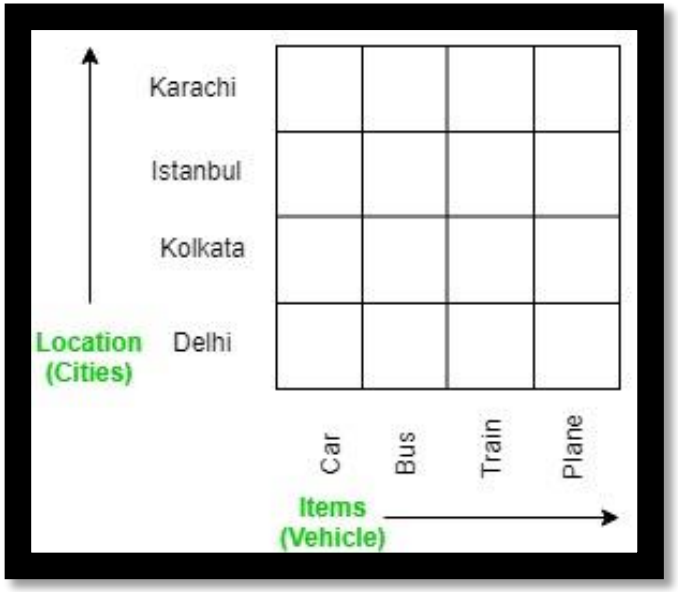


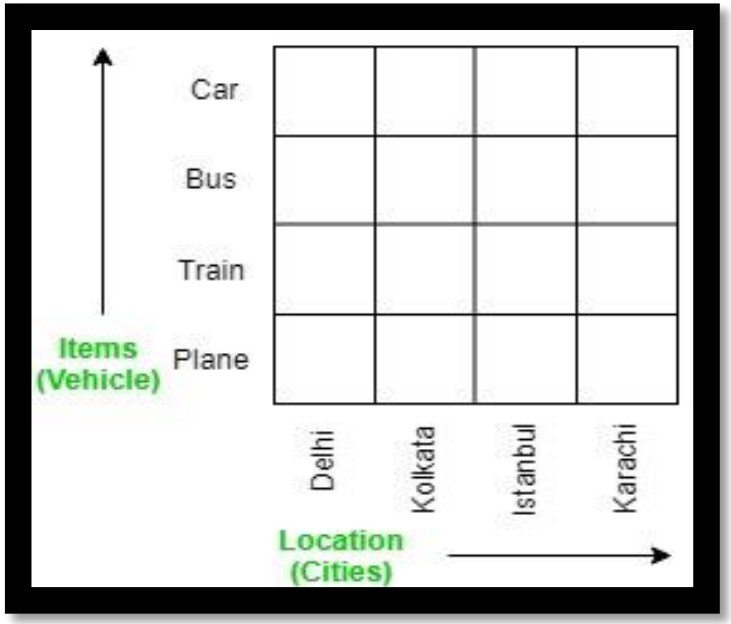
Операции OLAP.

Есть пять основных аналитических операций, которые можно выполнить с кубом OLAP:

Срез(slice): он выбирает одно измерение из куба OLAP, что приводит к созданию нового субкуба. В кубе, приведенном в разделе обзора, срез выполняется по измерению Time = «Q1».

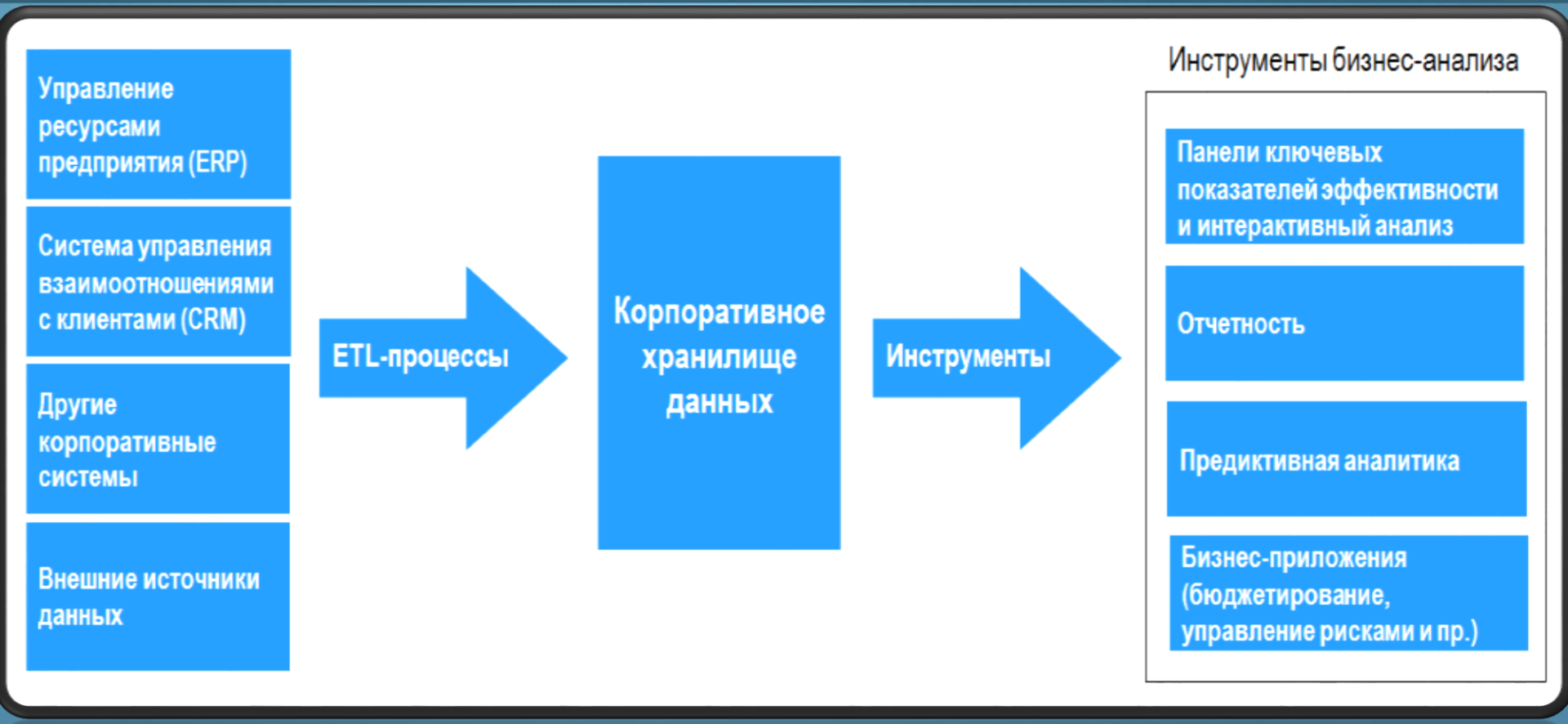
Операция поворота, вращает текущий вид, для получения нового вида представления. В подкубе, полученном после операции среза, выполнение операции поворота дает новое представление о нем.





	OLTP	OLAP
Характеристики	Обрабатывает большое количество мелких транзакций	Обрабатывает сложными запросами большие объемы данных
Типы запросов	Простые стандартизированные запросы	Сложные запросы
Операции	На основе команд INSERT, UPDATE, DELETE	Агрегирования данных для отчетности на основе команд SELECT
Время отклика	Миллисекунды	Секунды, минуты или часы в зависимости от объема данных для обработки
Дизайн	Специфические для отрасли, например, розничная торговля, производство или банковское дело.	По предмету, например по продажам, инвентарю или маркетингу.
Источник	Платежи, проводки	Агрегированные данные по транзакциям
Цель	Контролировать и выполнять важные бизнес-операции в режиме реального времени	Планировать, решать проблемы, поддерживать решения, обнаруживать скрытые идеи
Обновления данных	Короткие и быстрые обновления, инициированные пользователем	Данные периодически обновляются с помощью запланированных длительных пакетных заданий.
Требования к пространству	Обычно небольшой, если архивируются исторические данные	Обычно большой из-за агрегирования больших наборов данных
Резервное копирование и восстановление	Регулярное резервное копирование, необходимое для обеспечения непрерывности бизнеса и соответствия законодательным и корпоративным требованиями.	Потерянные данные могут быть подгружены из базы данных OLTP по мере необходимости вместо регулярного резервного копирования.
Продуктивность	Повышает продуктивность конечных пользователей	Повышает продуктивность бизнес-менеджеров, аналитиков данных и руководителей
Просмотр данных	Отображает повседневные бизнес-операции	Многомерное представление корпоративных данных
Примеры пользователей	Персонал, работающий с клиентами, клерки, онлайн-покупатели	Работники умственного труда, такие как аналитики данных, бизнес-аналитики и руководители
Дизайн базы данных	Для пущей эффективности базы данных нормализуются	Базы данных денормализированы при анализе

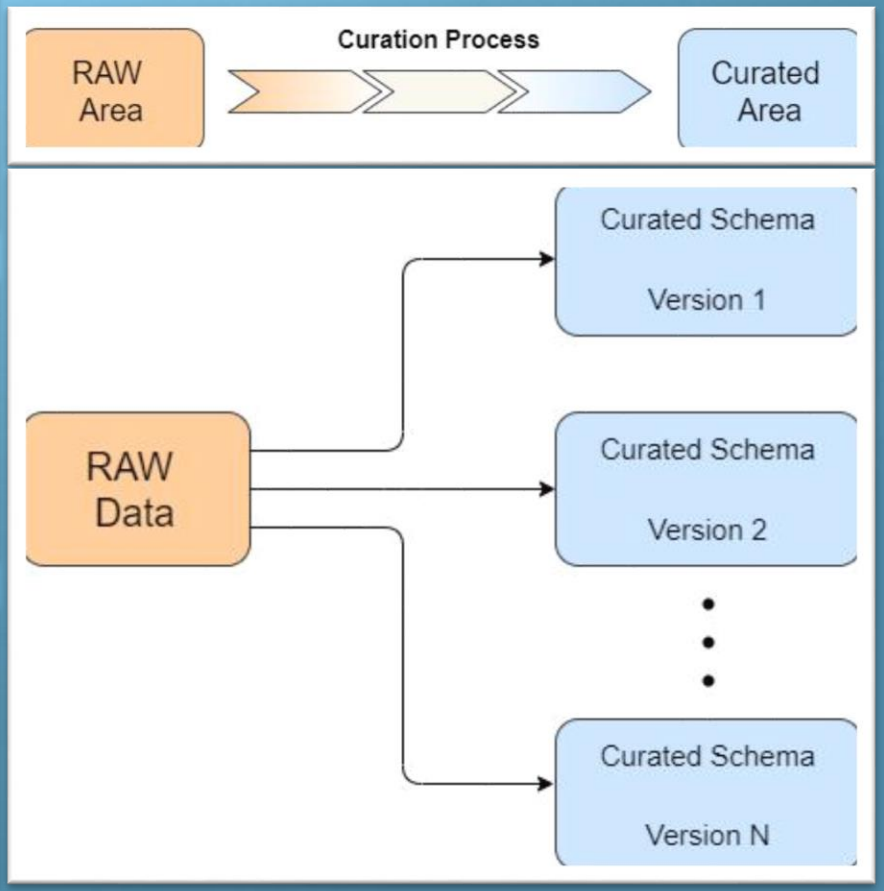
КОНЦЕПЦИЯ ХРАНИЛИЩА ДАННЫХ



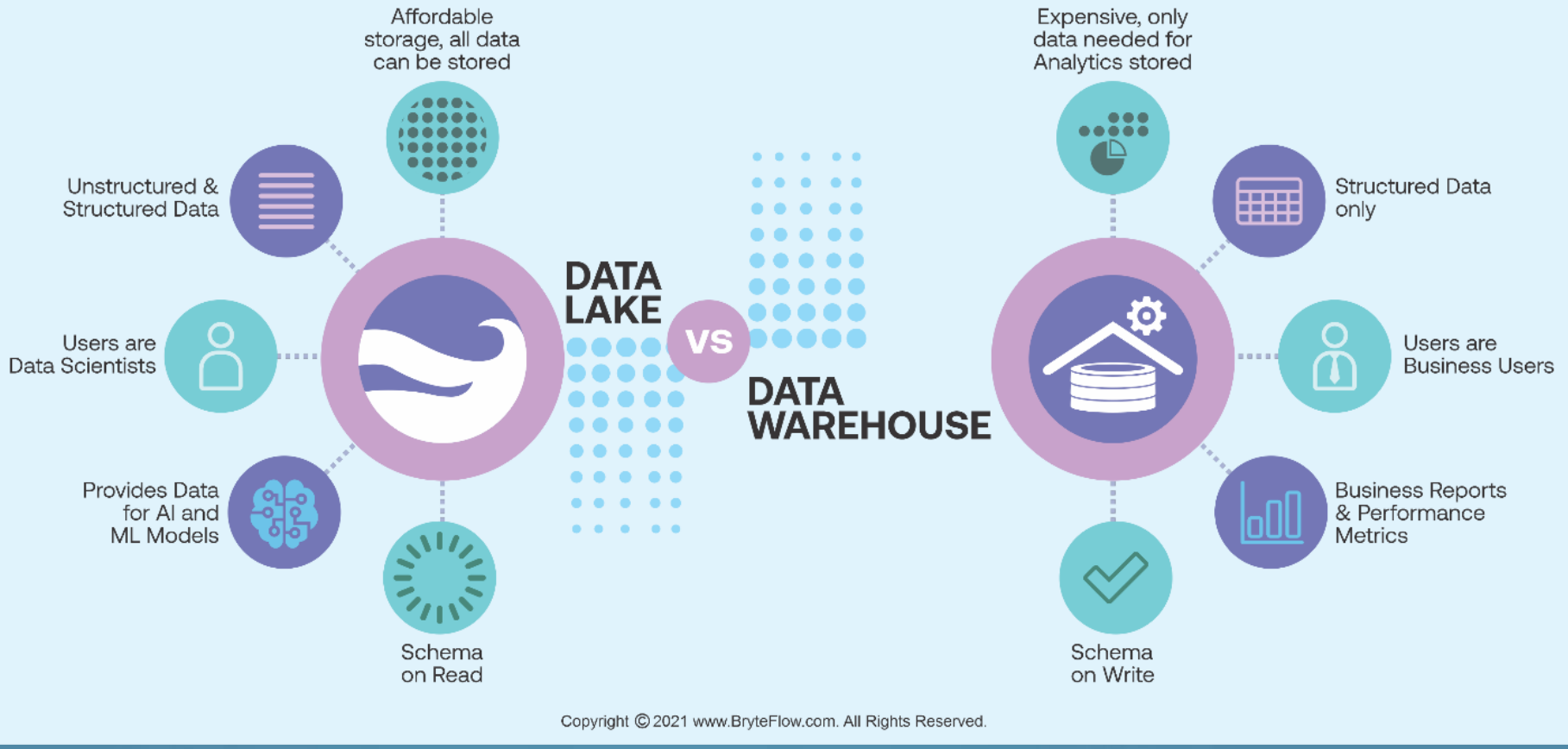
Techtarget.com : «Управление данными - это процесс приема, хранения, организации и обслуживания данных, созданных и/или накопленных организацией ».

Ральф Кимбалл (Ralph Kimball), один из авторов концепции хранилищ данных, описывал хранилище данных как "место, где люди могут получить доступ к своим данным" . Он же сформулировал и основные требования к хранилищам данных:

- Обеспечение высокой скорости получения данных из хранилища;
- поддержка внутренней непротиворечивости данных;
- возможность получения и сравнения так называемых срезов данных (slice and dice);
- наличие удобных утилит просмотра данных в хранилище;
- полнота и достоверность хранимых данных;
- поддержка качественного процесса пополнения данных ;

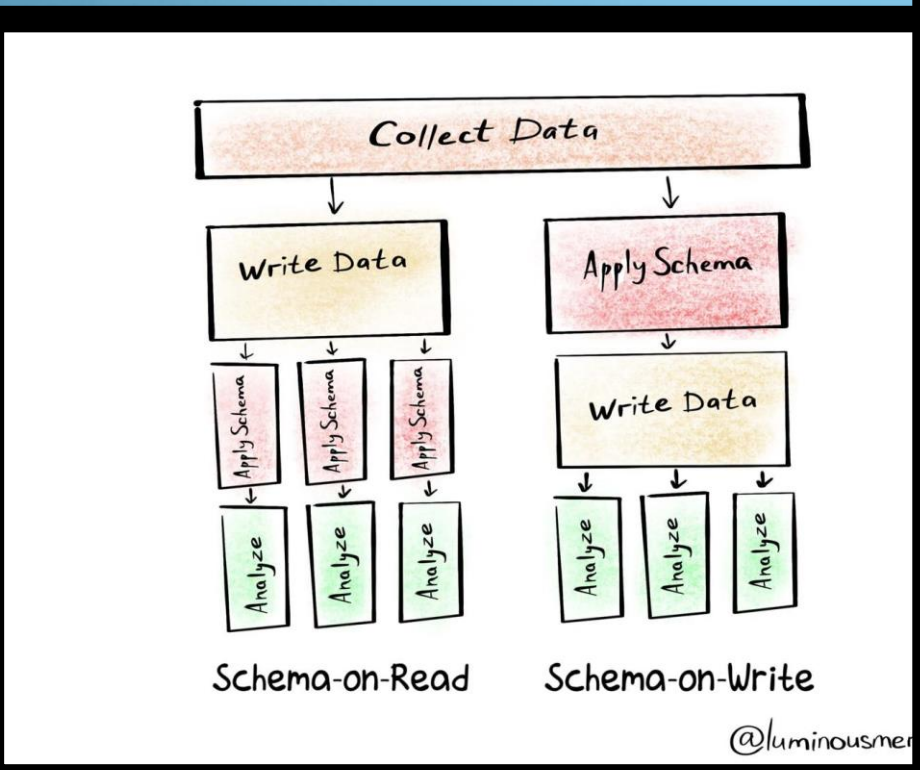


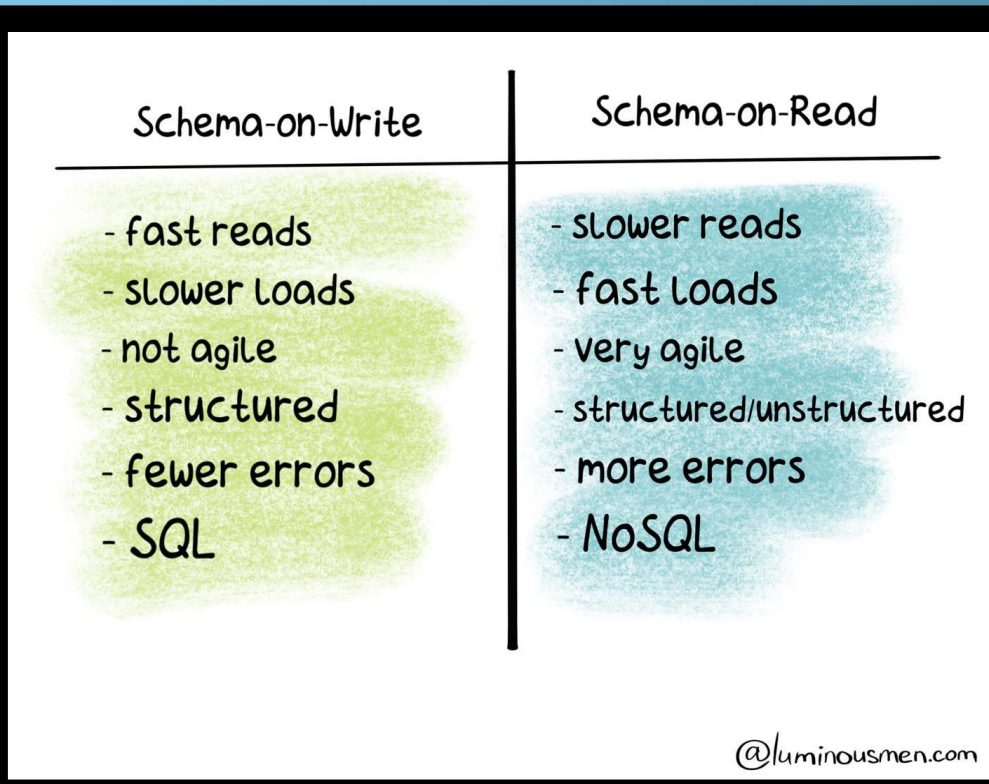
Озеро данных (Data Lake) представляет собой резервуар для хранения больших данных, который содержит огромное количество нерафинированной информации. Данные загружаются непосредственно в озеро данных, не проходя через уровень интеграции или уровень преобразования.



Импортированные данные могут быть структурированными, например, таблицы реляционной базы данных, полуструктурированными, как файлы CSV, JSON, Parquet, или неструктурированными, например PDF-файлами и изображениями. (Hadoop, Azure, Amazon S3)

Шаблон на запись (Schema-on-Write). Управленяет реляционной базой данных, включая создание схемы и таблицы, а также прием данных. Уловка 22 здесь заключается в том, что данные не могут быть загружены в таблицы без создания и настройки схем и таблиц. В противоположность этому, рабочая структура базы данных не может быть определена без понимания структуры данных, которые должны быть загружены в базу данных





Шаблон на чтение (Schema-on-Read). Если ваше озеро данных содержит данные, появляющиеся в реальном времени, с помощью конструкции schema-on-read новые поля будут добавляться в схему базы данных по мере необходимости и по мере загрузки данных.

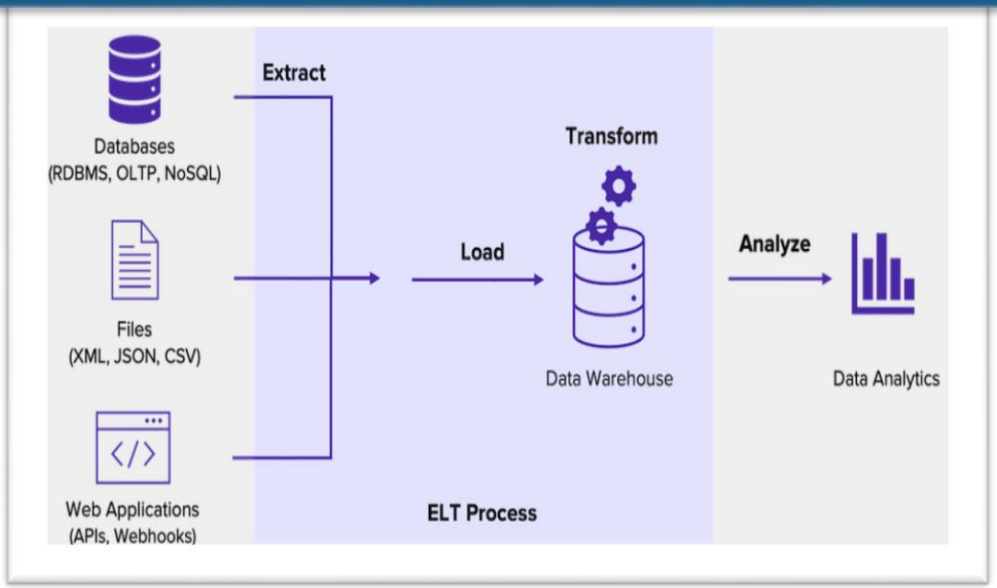
В инженерии данных существует два общих подхода к взаимодействию с данными. Первый подход - ETL (извлечение, преобразование, загрузка). Второй подход - ELT (извлечение, загрузка, преобразование).

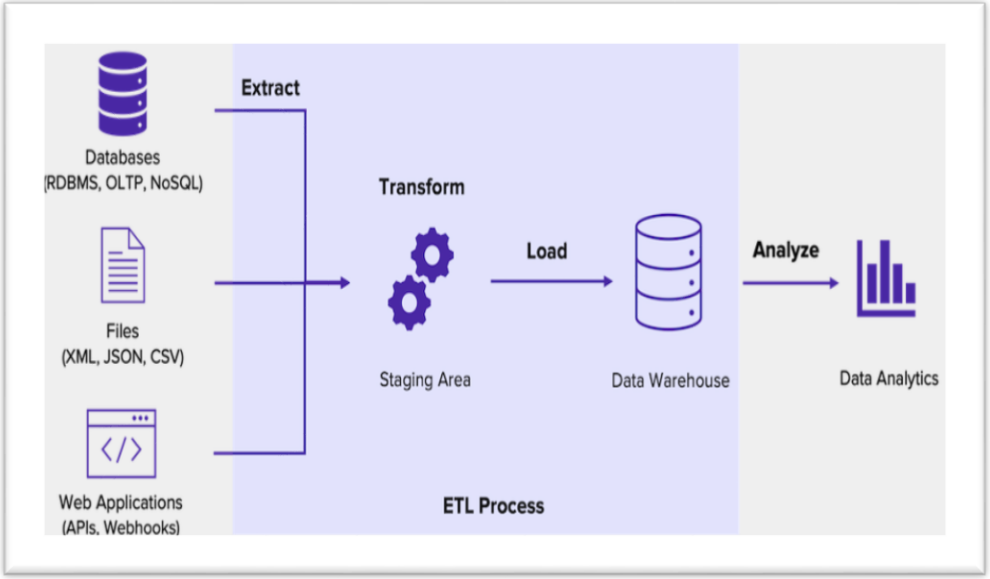
- **Извлечение данных (Extract)** из различных источников (пользовательские и системные логи, реляционные СУБД, внешние датасеты, например, из соцсетей и прочих веб-сайтов, Facebook Ads, Google Analytics, Yandex Metrics);

- **Преобразование (Transform)**, чтобы преобразовать сырые данные в готовый к анализу датасет, (скажем, необходимо сформировать сводную таблицу или провести сложный когортный анализ ваших пользователей), к информации применяются различные операции бизнес-логики — фильтрация, группировка и агрегирование;

- **Загрузка (Load)** — отправка обработанной информации в место конечного использования — озеро данных (Data Lake), СУБД, витрина данных, облачное приложение Amazon S3, дэшборды BI-системы Tableau и т. д. (Дашборд — это панель с визуализацией данных. Чаще всего это выглядит как иллюстрация важнейших метрик с инфографикой)

ОБСЛУЖИВАНИЕ ДАННЫХ ВКЛЮЧАЕТ В СЕБЯ ТАКИЕ ДЕЙСТВИЯ С ДАННЫМИ, КАК ПЕРЕМЕЩЕНИЕ, ИНТЕГРАЦИЯ, ОЧИСТКА, ОБОГАЩЕНИЕ И ETL-ПРОЦЕССЫ (EXTRACT, TRANSFORM, LOAD).





- **Процессы ETL** (Extract, Transform, Load) являются неотъемлемой частью современных систем бизнес-аналитики (BI, Business Intelligence) и используются для интеграции множества корпоративных информационных систем с целью унификации и анализа хранимых в них данных.
- Можно сказать, что сегодня ETL — это обязательный компонент корпоративной инфраструктуры на базе технологий Big Data, когда исходные («сырые») данные превращаются в информацию, пригодную для бизнес-анализа. ETL

ТЕХСТЕК ETL И ELT - 10 КЛЮЧЕВЫХ ОТЛИЧИЙ:

	ETL	ELT
1) ПОДДЕРЖКА ХРАНИЛИЩА ДАННЫХ	Да, ETL — это традиционный процесс преобразования и интеграции структурированных или реляционных данных в облачное или локальное хранилище данных.	Да, ELT — это современный процесс преобразования и интеграции структурированных или неструктурированных данных в облачное хранилище данных.
2) ПОДДЕРЖКА DATA LAKE / MART / LAKEHOUSE	Нет, ETL не подходит для озер данных, витрин данных или хранилищ данных.	Да, процесс ELT предназначен для обеспечения конвейера данных для озер данных, витрин данных или хранилищ данных.
3) РАЗМЕР / ТИП НАБОРА ДАННЫХ	ETL лучше всего подходит для обработки небольших реляционных наборов данных, которые требуют сложных преобразований и заранее определены как имеющие отношение к целям анализа.	ELT может обрабатывать данные любого размера и типа и хорошо подходит для обработки как структурированных, так и неструктурированных больших данных. Поскольку загружен весь набор данных, аналитики могут в любой момент выбрать, какие данные преобразовать и использовать для анализа.

4) РЕАЛИЗАЦИЯ	**Процесс ETL существует уже несколько десятилетий, и существует развитая экосистема инструментов ETL и экспертов, готовых помочь с внедрением.**	**Процесс ELT — это новый подход, и экосистема инструментов и экспертов, необходимых для его реализации, все еще растет.**
5) ПРЕОБРАЗОВАНИЕ	**В процессе ETL преобразование данных выполняется в промежуточной области за пределами хранилища данных, и все данные должны быть преобразованы перед загрузкой. В результате преобразование больших наборов данных может занять много времени, но анализ может выполняться сразу после завершения процесса ETL.**	**В процессе ELT преобразование данных выполняется по мере необходимости в самой целевой системе. В результате этап преобразования занимает мало времени, но может замедлить процессы запросов и анализа, если нет достаточной вычислительной мощности.**
6. ЗАГРУЗКА	**Шаг загрузки ETL требует, чтобы данные были загружены в промежуточную область перед загрузкой в целевую систему. Этот многоэтапный процесс занимает больше времени, чем процесс ELT.**	**В ELT полный набор данных загружается непосредственно в целевую систему. Поскольку существует только один шаг, и он выполняется только один раз, загрузка в процессе ELT происходит быстрее, чем в ETL.**

7) ОБСЛУЖИВАНИЕ / ПРОСТОТА ИСПОЛЬЗОВАНИЯ	**Процессы ETL, в которых задействован локальный сервер, требуют частого обслуживания ИТ-специалистами с учетом их фиксированных таблиц, фиксированных сроков и необходимости многократно выбирать данные для загрузки и преобразования. Новые автоматизированные облачные решения ETL не требуют значительного обслуживания.**	**Процесс ELT обычно требует минимальных затрат на обслуживание, учитывая, что все данные всегда доступны, а процесс преобразования обычно автоматизирован и основан на облаке.**
8) СТОИМОСТЬ	**ETL может быть слишком дорогостоящим для многих малых и средних предприятий.**	**ELT извлекает выгоду из надежной экосистемы облачных платформ, которые предлагают гораздо более низкие затраты и различные варианты планов для хранения и обработки данных.**
9) АППАРАТНОЕ ОБЕСПЕЧЕНИЕ	**Традиционный локальный процесс ETL требует дорогостоящего оборудования. Новые облачные решения ETL не требуют оборудования.**	**Учитывая, что процесс ELT изначально основан на облаке, дополнительное оборудование не требуется.**
10) СООТВЕТСТВИЕ	**ETL лучше подходит для соответствия стандартам GDPR, HIPAA и CCPA, учитывая, что пользователи могут опускать любые конфиденциальные данные перед загрузкой в целевую систему.**	**ELT несет в себе больший риск раскрытия личных данных и несоблюдения стандартов GDPR, HIPAA и CCPA, поскольку все данные загружаются в целевую систему.**

СТАДИИ СТАНОВЛЕНИЯ DATA LAKE

Уровень зрелости управления	Состояние и характер данных	Состояние Data Lake
1. Начальный	Данные дублируются или частично отсутствуют, представлены в разных форматах и системах, не связаны между собой, велика доля ручной обработки данных	Локальное хранилище данных без определенного порядка автоматизированной обработки
2. Управляемый	Информация достаточно успешно обрабатывается автоматически в пределах одного подразделения, но не интегрирована с другими корпоративными процессами и структурами (отделами, филиалами и пр.)	Лужа или болото данных
3. Определенный	Обмен данными между различными процессами, системами и структурами предприятия частично автоматизирован, имеется единый каталог корпоративных данных	Озеро данных
4. Управляемый на основе количественных данных	Синхронизация данных между различными процессами, системами и структурами предприятия автоматизирована не полностью, часть процедур запускается по требованию или вручную	Управляемое озеро данных
5. Оптимизируемый	Процедуры автоматизированного появления, обновления, обмена и синхронизации данных между различными процессами, системами и структурами предприятия отлажены и успешно работают	Самоорганизующееся озеро данных

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ОЗЕРА ДАННЫХ

Здравоохранение. Из-за большого количества неструктурированных данных в сфере здравоохранения (например, записей врачей, клинических данных и т. Д.) И необходимости получения информации в реальном времени использование озер данных позволяет получить доступ к структурированным и неструктурированным данным, которые, как оказалось, являются лучше подходит для медицинских компаний.

Образование. Сбор данных об оценках учащихся, посещаемости и т. Д. Может не только помочь учащимся улучшить их послужной список, но также может помочь предсказать потенциальные проблемы до того, как они возникнут.

Транспорт. Озера данных - отличный источник информации из-за их способности делать прогнозы. В транспортной отрасли прогнозы могут помочь компаниям сократить расходы и улучшить профилактическое обслуживание

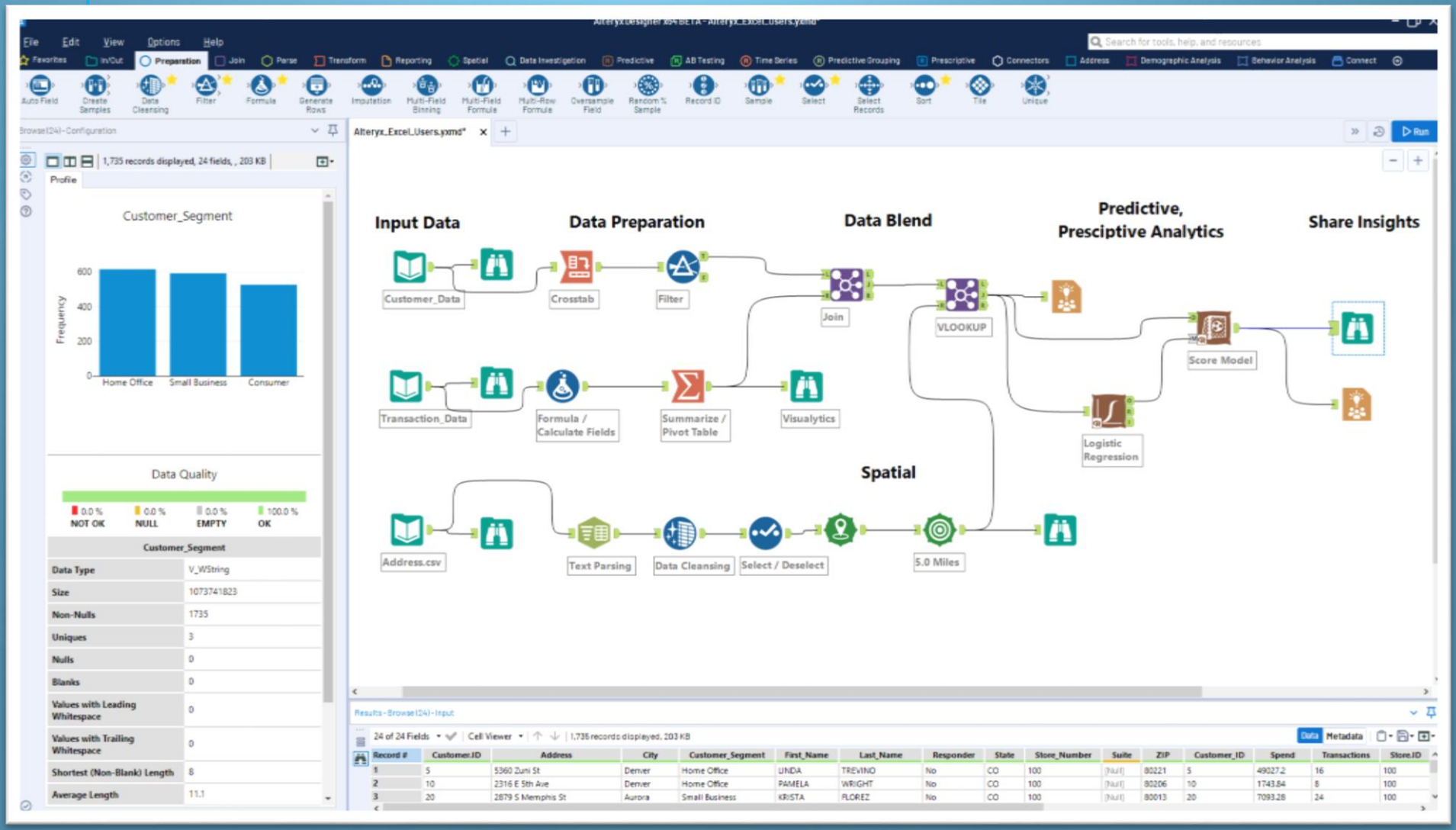
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ХРАНИЛИЩА ДАННЫХ

Банковское дело и финансы. Хранилище данных часто является лучшей моделью хранения для этих секторов, поскольку они обеспечивают структурированный доступ для всей компании, а не для одного специалиста по данным.

Государственный сектор. Помогает агентствам вести и анализировать налоговую отчетность, политику в области здравоохранения и т. Д., Создавая как индивидуальные профили, так и групповые записи.

Индустрия туризма. Эта отрасль использует хранилища данных для разработки ориентированных на клиентов, на основе их отзывов и моделей поездок, продвижений и рекламных кампаний,. Она также используют DW для выполнения повседневных операций.

ПРОПРИЕТАРНЫЕ ETL-ИНСТРУМЕНТЫ.
Pentaho Data Integration и Talend Open Studio, Alteryx Designer.



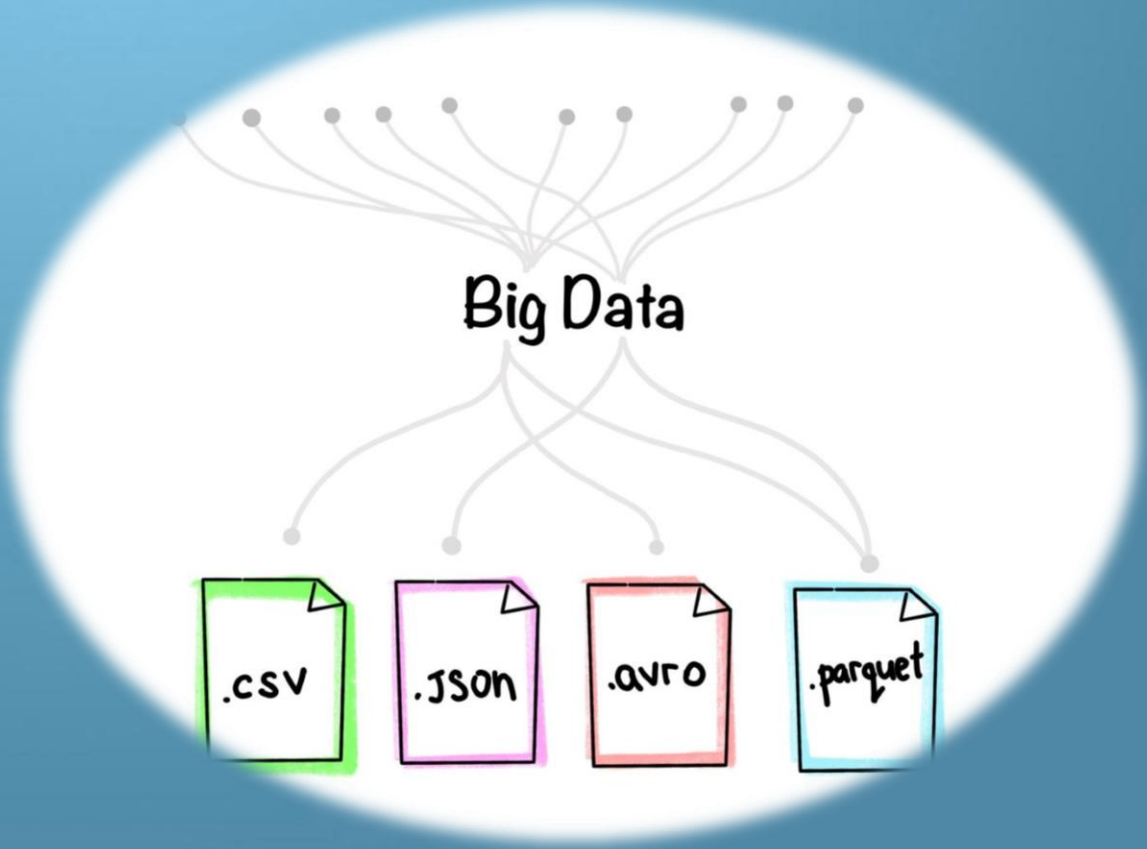
OPEN-SOURCE ETL-ИНСТРУМЕНТЫ.
Airbnb’s Airflow, Spotify’s Luigi

ФОРМАТЫ ФАЙЛОВ ХРАНЕНИЯ BIG DATA

Использование усовершенствованного формата файла призвано привнести в систему следующие преимущества:

1. Время чтения уменьшается.
2. Время записи сокращается.
3. Файлы могут быть разделены, не надо считывать весь файл для получения меньшего его подраздела.
4. Возможность поддержки эволюции редактирования схемы, и схема может быть изменена по запросу в зависимости от изменения потребностей системы.
5. Имеются кодеки для обеспечения возможности сжатия файлов без потери преимущества базового формата.

В простейшем случае это Открытый текст:
CSV, XML, JSON, JSONB, YAM, BLOB итд



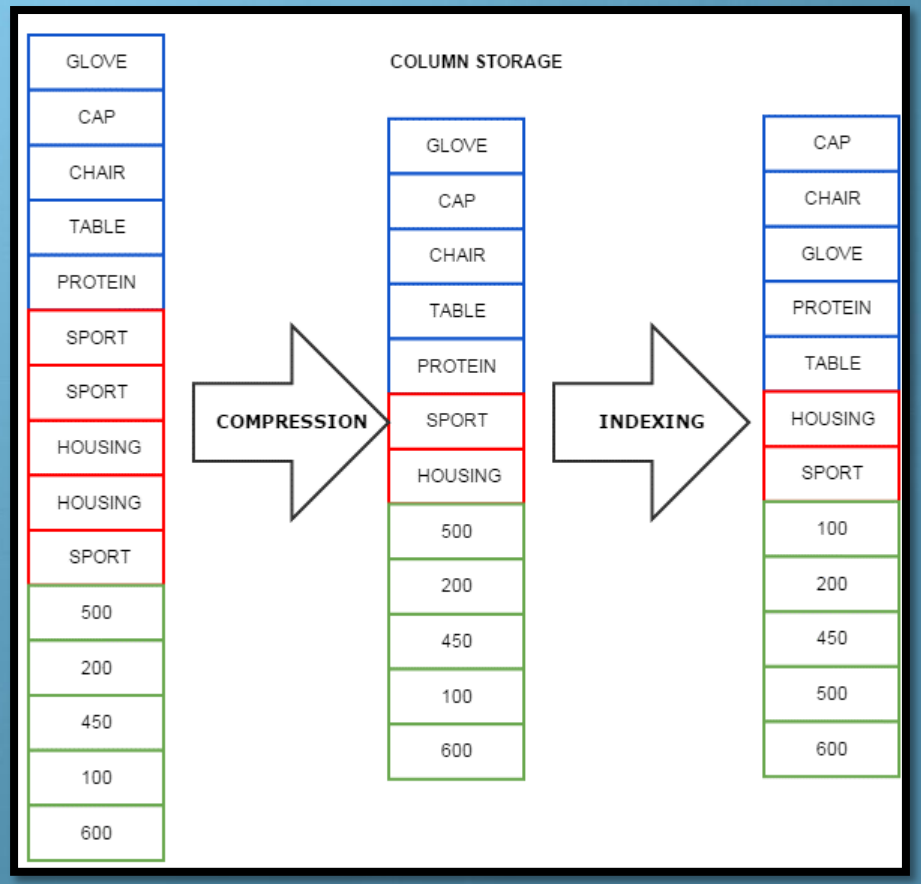
ФОРМАТЫ ФАЙЛОВ ХРАНЕНИЯ BIG DATA

LOGICAL TABLE STRUCTURE

MATERIAL	CATEGORY	REVENUE (EUR)
GLOVE	SPORT	500
CAP	SPORT	200
CHAIR	HOUSING	450
TABLE	HOUSING	100
PROTEIN	SPORT	600

ROW STORAGE

GLOVE
SPORT
500
CAP
SPORT
200
CHAIR
HOUSING
450
TABLE
HOUSING
100
PROTEIN
SPORT
600



ТИПА ФОРМАТОВ ДЛЯ BIG DATA ФАЙЛОВ
линейные (строковые) и колоночные (столбцовые).

Row-oriented: rows stored sequentially in a file

Key	Fname	Lname	State	Zip	Phone	Age	Sales
1	Bugs	Bunny	NY	11217	(123) 938-3235	34	100
2	Yosemite	Sam	CA	95389	(234) 375-6572	52	500
3	Daffy	Duck	NY	10013	(345) 227-1810	35	200
4	Elmer	Fudd	CA	04578	(456) 882-7323	43	10
5	Witch	Hazel	CA	01970	(567) 744-0991	57	250

Column-oriented: each column is stored in a separate file

Each column for a given row is at the same offset.

Key
1
2
3
4
5

Fname
Bugs
Yosemite
Daffy
Elmer
Witch

Lname
Bunny
Sam
Duck
Fudd
Hazel

State
NY
CA
NY
CA
CA

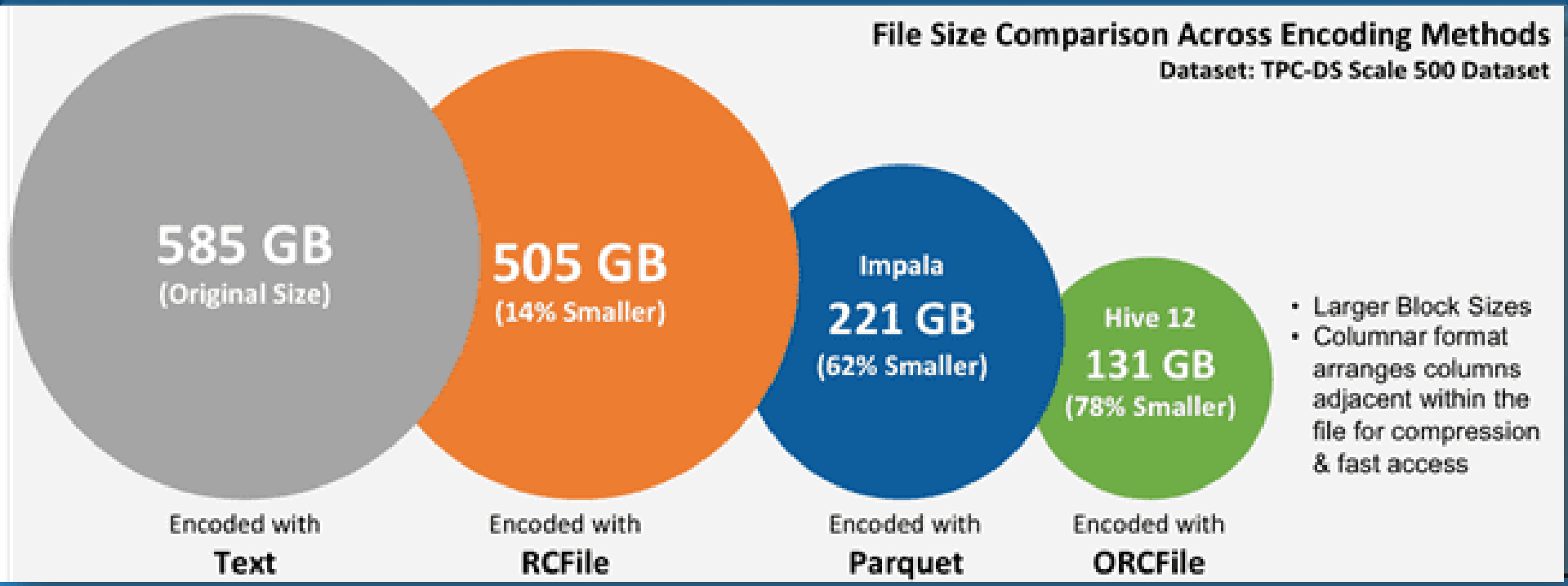
Zip
11217
95389
10013
04578
01970

Phone
(123) 938-3235
(234) 375-6572
(345) 227-1810
(456) 882-7323
(567) 744-0991

Age
34
52
35
43
57

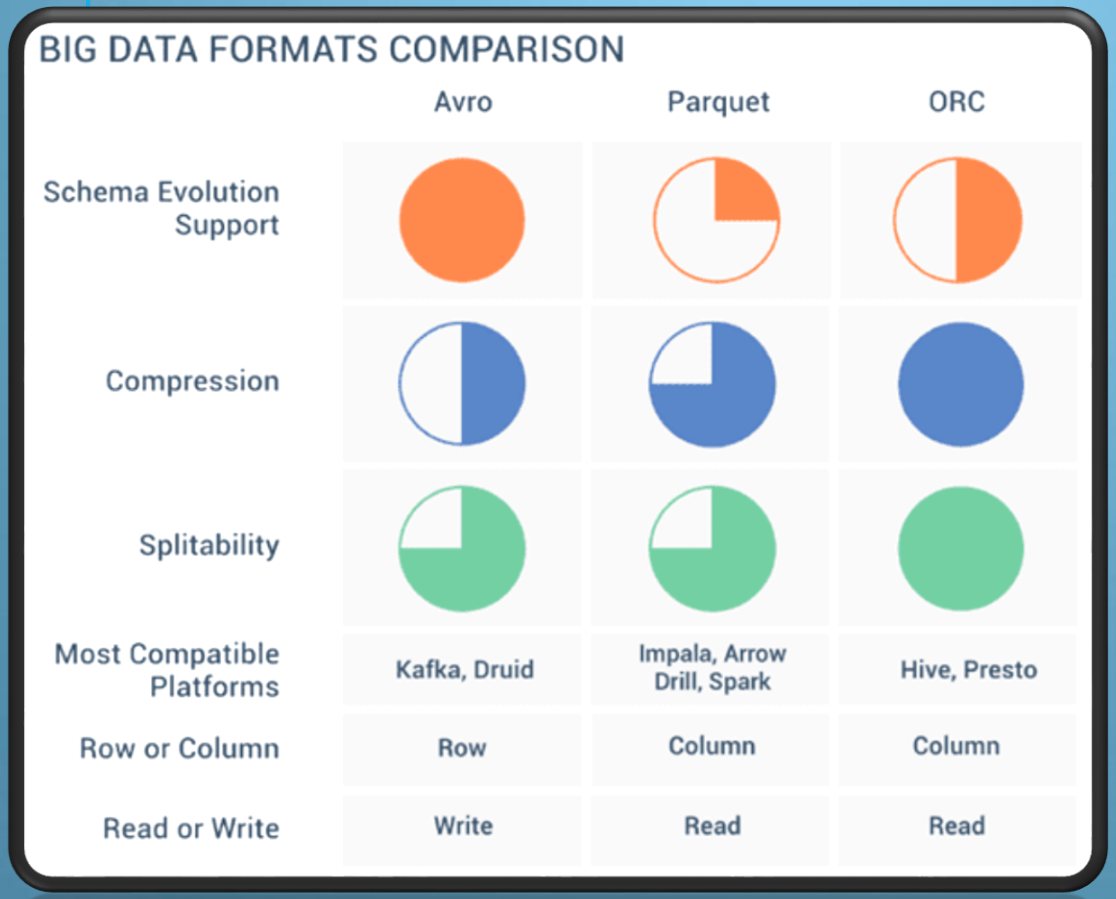
Sales
100
500
200
10
250

ОСНОВНЫЕ РАЗЛИЧИЯ МЕЖДУ ХРАНИЛИЩЕМ СТРОК И ХРАНИЛИЩЕМ СТОЛБЦОВ



Свойство	Хранилище строк	Хранилище столбцов	Причина
Использование памяти	Выше	Ниже	Сжатие
Транзакции	Быстрее	Медленнее	Изменения требуют обновления нескольких столбцовых таблиц
Аналитика	Медленнее, даже если индексировать	Быстрее	Меньший набор данных для сканирования, присущий индексации

ПРИМЕР ФОРМАТОВ ФАЙЛОВ BIGDATA



Apache Avro — наиболее популярная схема и система сериализации данных

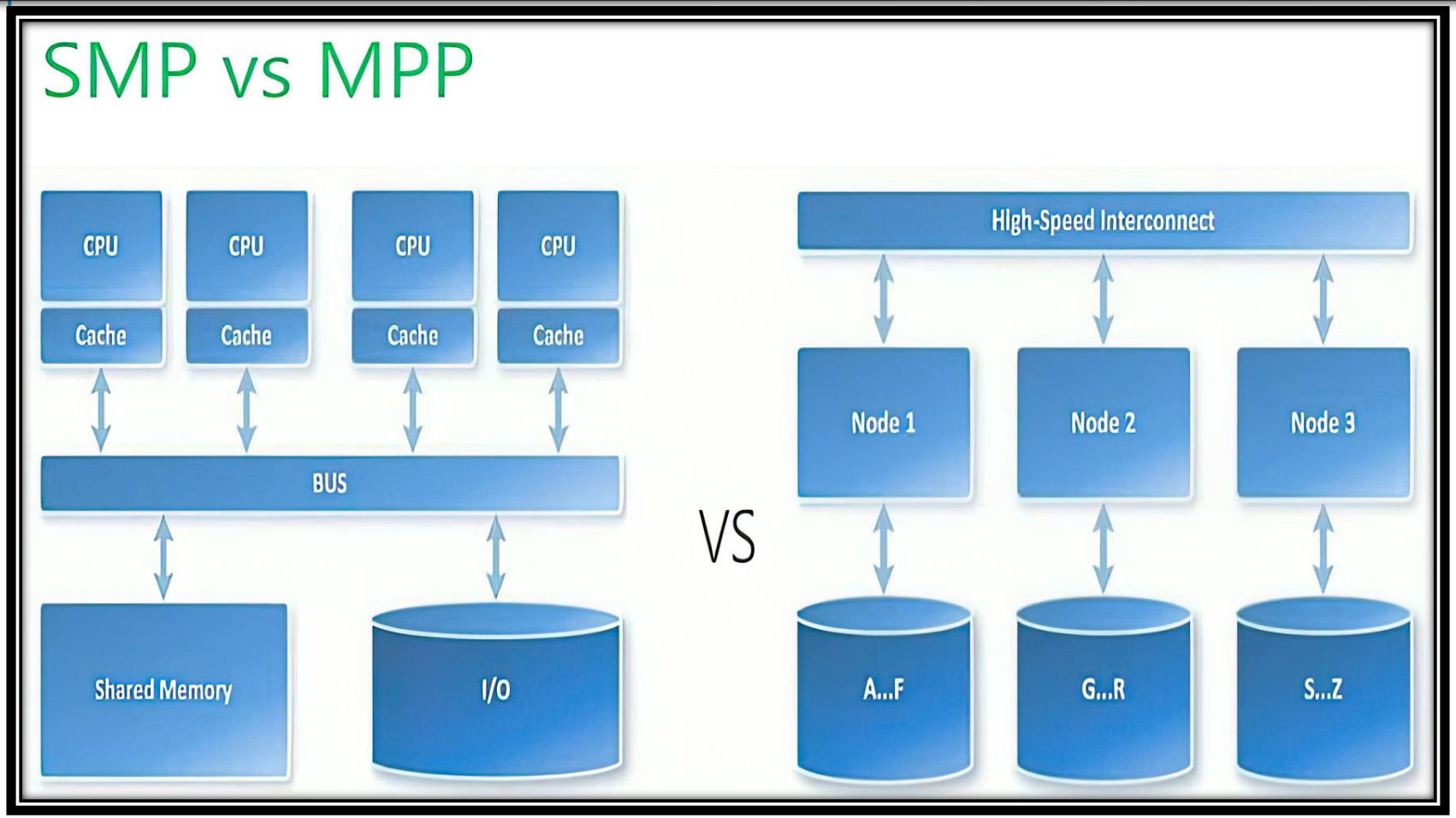
Apache Parquet— пзначально был разработан в Twitter, это бинарный, колоночно-ориентированный формат хранения Big Data.

ORC (Optimized Row Columnar)— это колоночно-ориентированный (столбцовый) формат хранения Big Data. Выпущен в феврале 2013 года Hortonworks и Facebook.

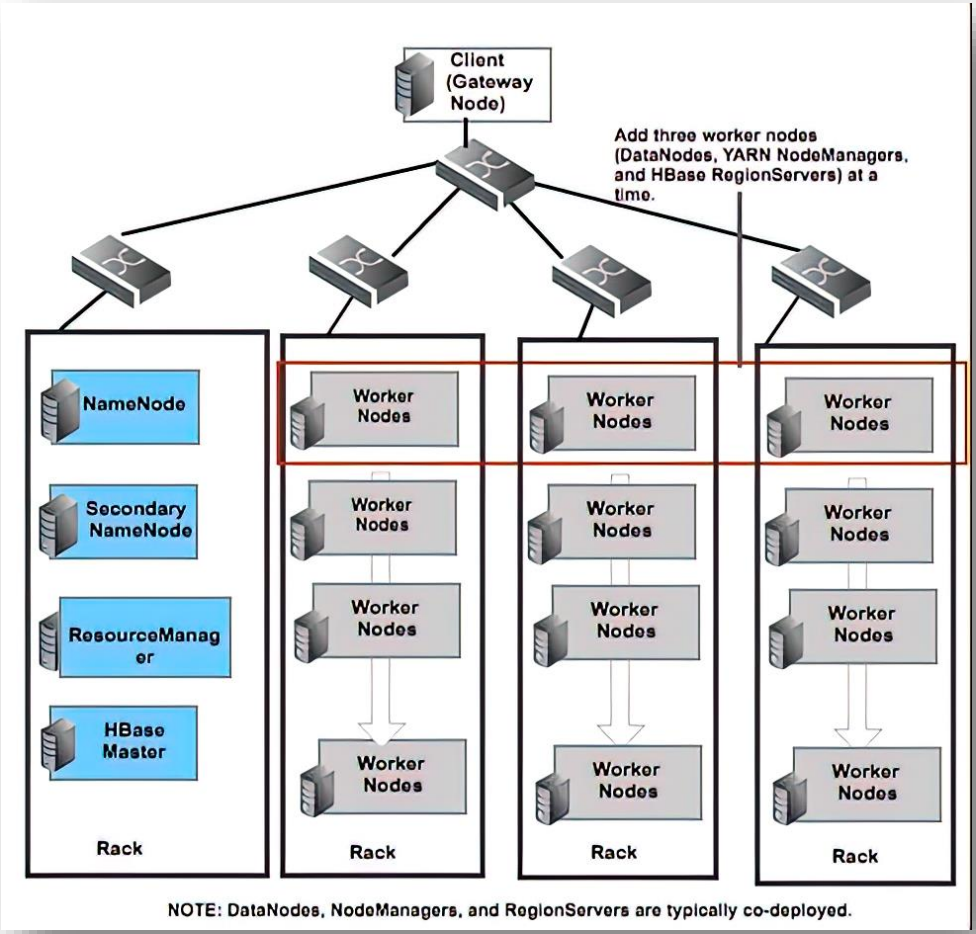
РАСПРЕДЕЛЕННАЯ (ПАРАЛЛЕЛЬНАЯ) ОБРАБОТКА ДАННЫХ

- **Hadoop:** Hadoop стартовал как проект Yahoo в 2006 году, но впоследствии стал опенсорсной надстройкой над Apache.
- Apache — это программное обеспечение с открытым исходным кодом, веб-сервер, который обеспечивает работу около 46% сайтов по всему миру.

Система симметричной многопроцессорной обработки (SMP) — это вычислительная архитектура, в которой все процессоры совместно используют одну и ту же операционную систему, память, дисковое хранилище и подключены через системную шину. Все традиционные продукты SQL Server, включая SQL Server 2005, 2008, 2012 и т. Д. используют архитектуру SMP.



Система с массовой параллельной обработкой (MPP), с другой стороны, использует подход «без общего доступа». В этой архитектуре каждый процессор имеет свой собственный набор ресурсов (память, дисковое хранилище, операционная система), и каждый процессор полностью независим и изолирован от других процессоров. Как вы можете догадаться, в этой архитектуре нет единой точки разногласий, и, следовательно, она может масштабироваться.



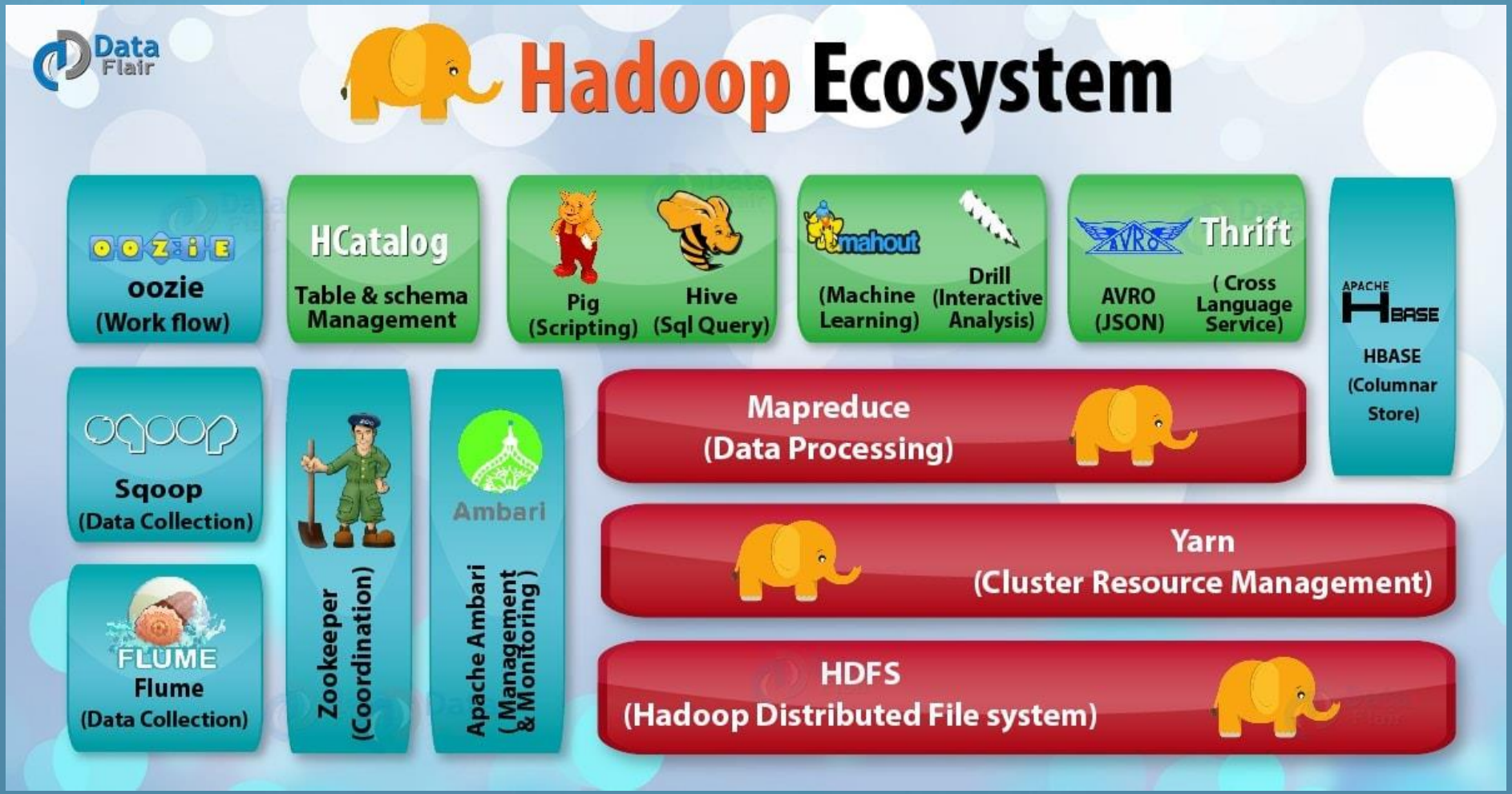
ПРИМЕРНО ПОЛПРОЦЕНТА ОТ ОБЩЕЙ ЕМКОСТИ AZURE.

Клуб exabyte: путь LinkedIn к масштабированию распределенной файловой системы Hadoop https://engineering.linkedin.com/blog/2021/the-exabyte-club--linkedin-s-journey-of-scaling-the-hadoop-distr

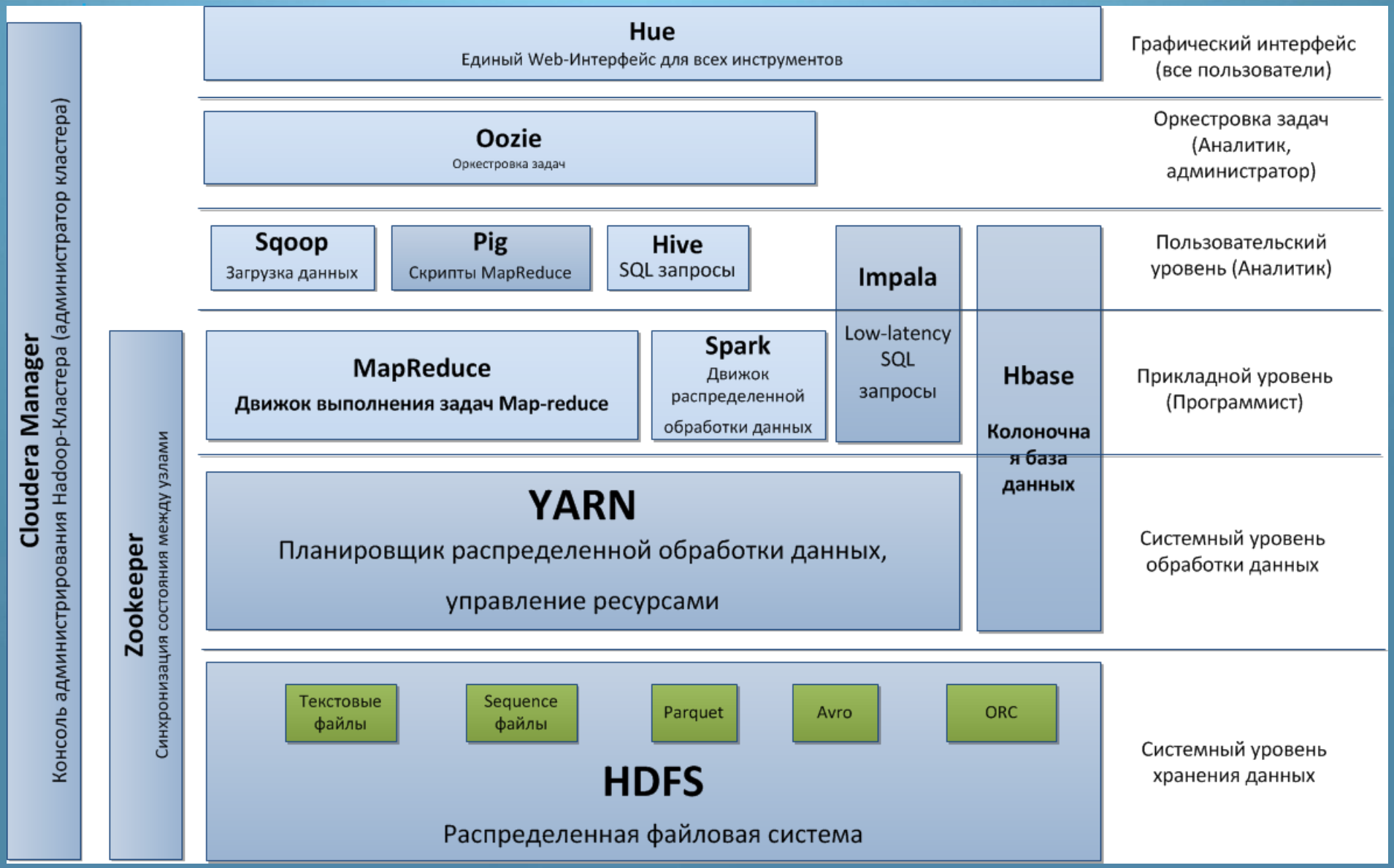
Самый большой кластер Hadoop на 2020 год - это LinkedIn с 20000 узлами и 500 ПБ емкости хранения.

(в коммерческой организации, не в агентстве безопасности крупной страны)

HADOOP 2. Составные части



HADOOP 2. Функциональная компоновка

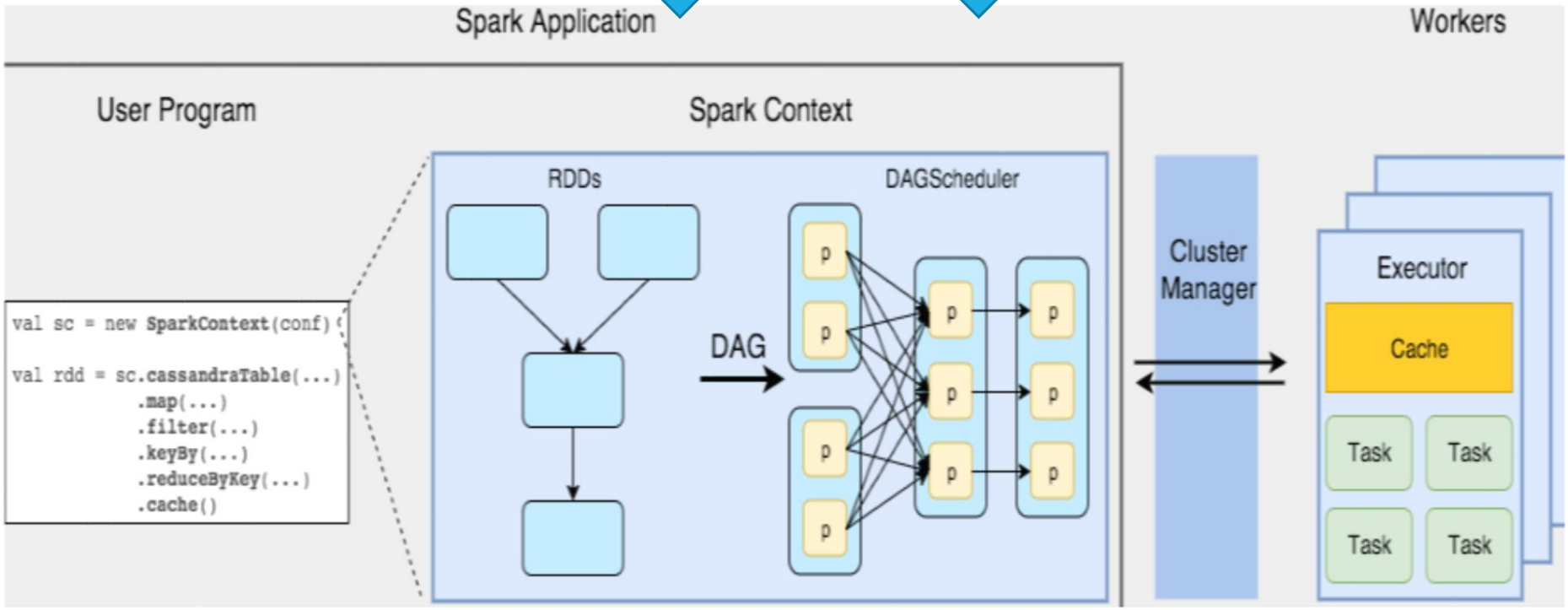


Дистрибутив	Общие компоненты	Файловая система	Управление кластером, координация, планирование	Управление интеграцией и потоками данных	Обеспечение безопасности	SQL СУБД	NoSQL СУБД	Потоковая обработка данных	Машинное обучение	Брокер сообщений
CLOUDERA	Hadoop Common, MapReduse, Yarn, Tez,полнотекстовый поиск Solr, язык запросов к слабоструктурированным данным Pig	HDFS	Cloudera Manager	Sqoop, Flume	Cloudera Navigator Encrypt, Sentry, RecordService	Hive, Impala,	Hbase	Spark Streaming	Mahout	Kafka
HORTONWORKS	Hadoop Common, MapReduse, Yarn, Tez,полнотекстовый поиск Solr, язык запросов к слабоструктурированным данным Pig	HDFS	Oozie, ZooKeeper, Ambari	Sqoop, Flume, Falcon, NFC, WebHDFS	Kerberos, Ranger, Knox	Hive, HCatalog,	HBase, Accumlo,	Storm	MLLib	Kafka
MAPR	Hadoop Common, MapReduse, Yarn, Tez,полнотекстовый поиск Solr, язык запросов к слабоструктурированным данным Pig	MapR-FS	Oozie, ZooKeeper, Sahara	Sqoop, Flume, Hue, HttpFS	Kerberos, MapR Native Security	Drill, Hive, Impala, Spark SQL	HBase	Storm	Mahout, GraphX, MLLib	MapR Event Store
ARENADATA	Hadoop Common, MapReduse, Yarn, Tez,полнотекстовый поиск Solr, язык запросов к слабоструктурированным данным Pig	HDFS	Oozie, ZooKeeper, Ambari	Sqoop, Flume, NFC, WebHDFS,	Atlas, Ranger, Knox	Hive	HBase	NiFi, NFC, Flink	Mahout, Giraph, MLLib	Kafka

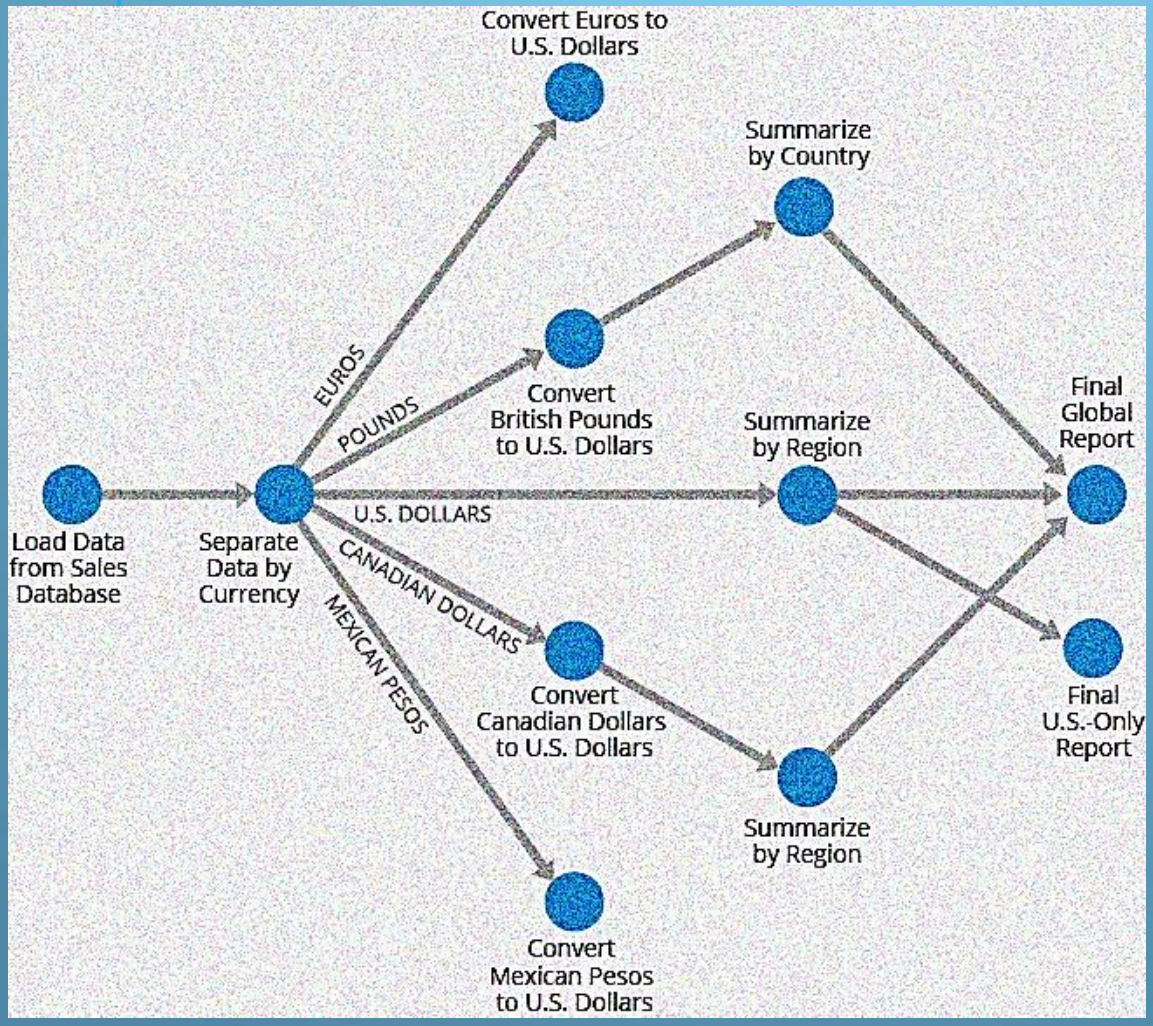
APACHE SPARK

RDD – распределенные контейнеры данных

Процессы обработки данных, или пайплайны, в Airflow описываются при помощи DAG – Direct Acyclic Graph Направленный ациклический граф задач



APACHE SPARK

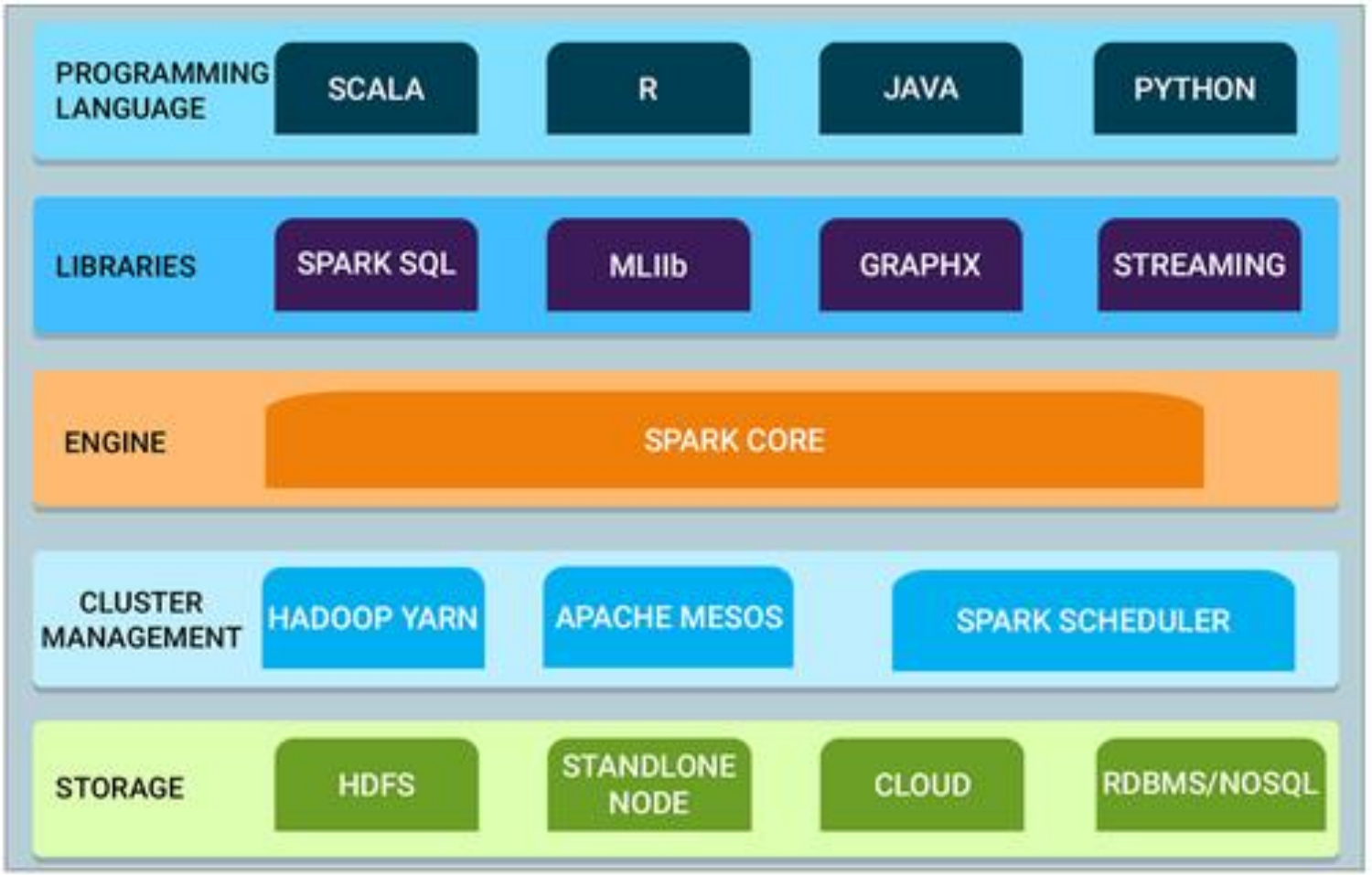


DAG в основном описывает, как мы хотим выполнять наш рабочий процесс.
Работа группы DAG заключается в том, чтобы убедиться, что все, что они делают:

- происходит в нужное время,
- в правильном порядке
- с правильной обработкой любых неожиданных проблем.

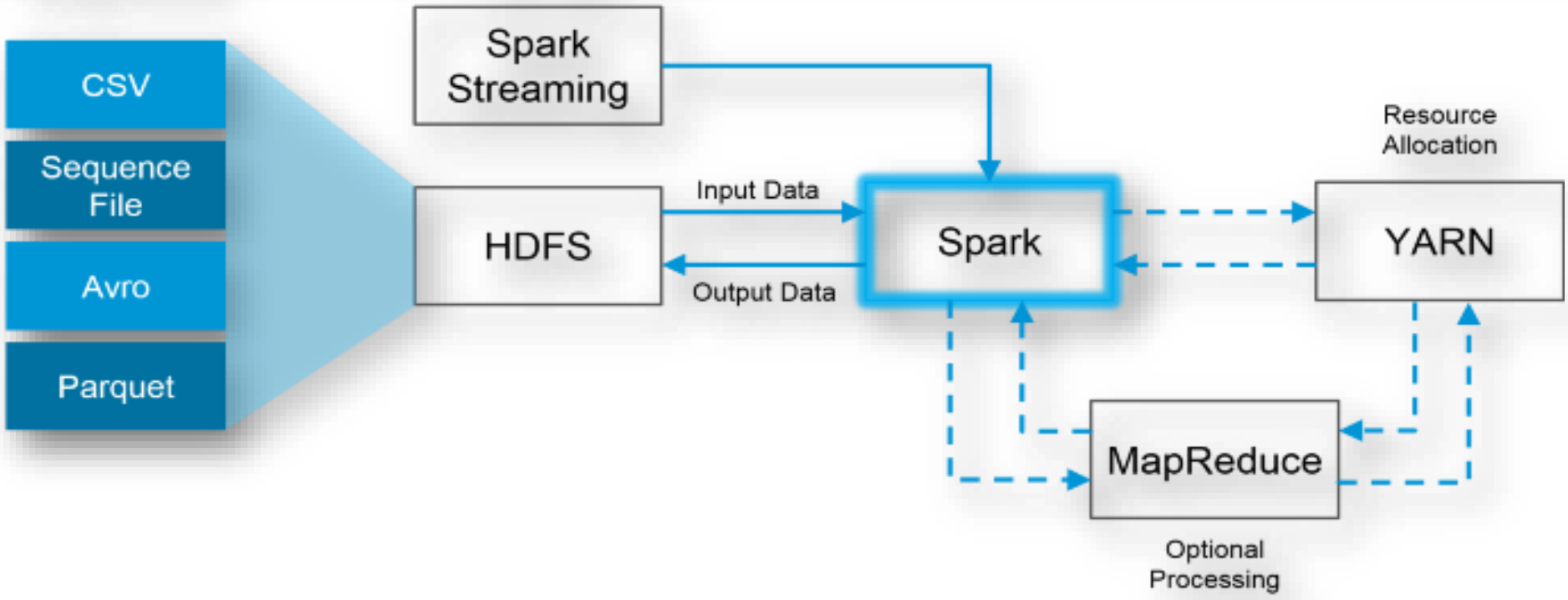
На уровне кода задачи могут представлять собой Python-функции или Bash-скрипты.

Новый проект, первоначально разработанный в 2012 году в AMPLab в Калифорнийском университете в Беркли. Это надстройка над Apache, ориентированная на параллельную обработку данных в кластере.



В то время как Hadoop читает и записывает файлы в HDFS, Spark обрабатывает данные в ОЗУ, используя концепцию, известную как Resilient Distributed Dataset (RDD), устойчивый распределенный набор данных.

Spark может работать как в автономном режиме, с кластером Hadoop, выступающим в качестве источника данных, так и в сочетании с Mesos. В последнем сценарии мастер Mesos подменяет мастер-планировщик Spark или YARN. Spark построен на основе Spark Core, механизма, который управляет планированием, оптимизацией и абстракцией RDD, а также подключает Spark к нужной файловой системе (HDFS, S3, RDBMS или Elasticsearch).

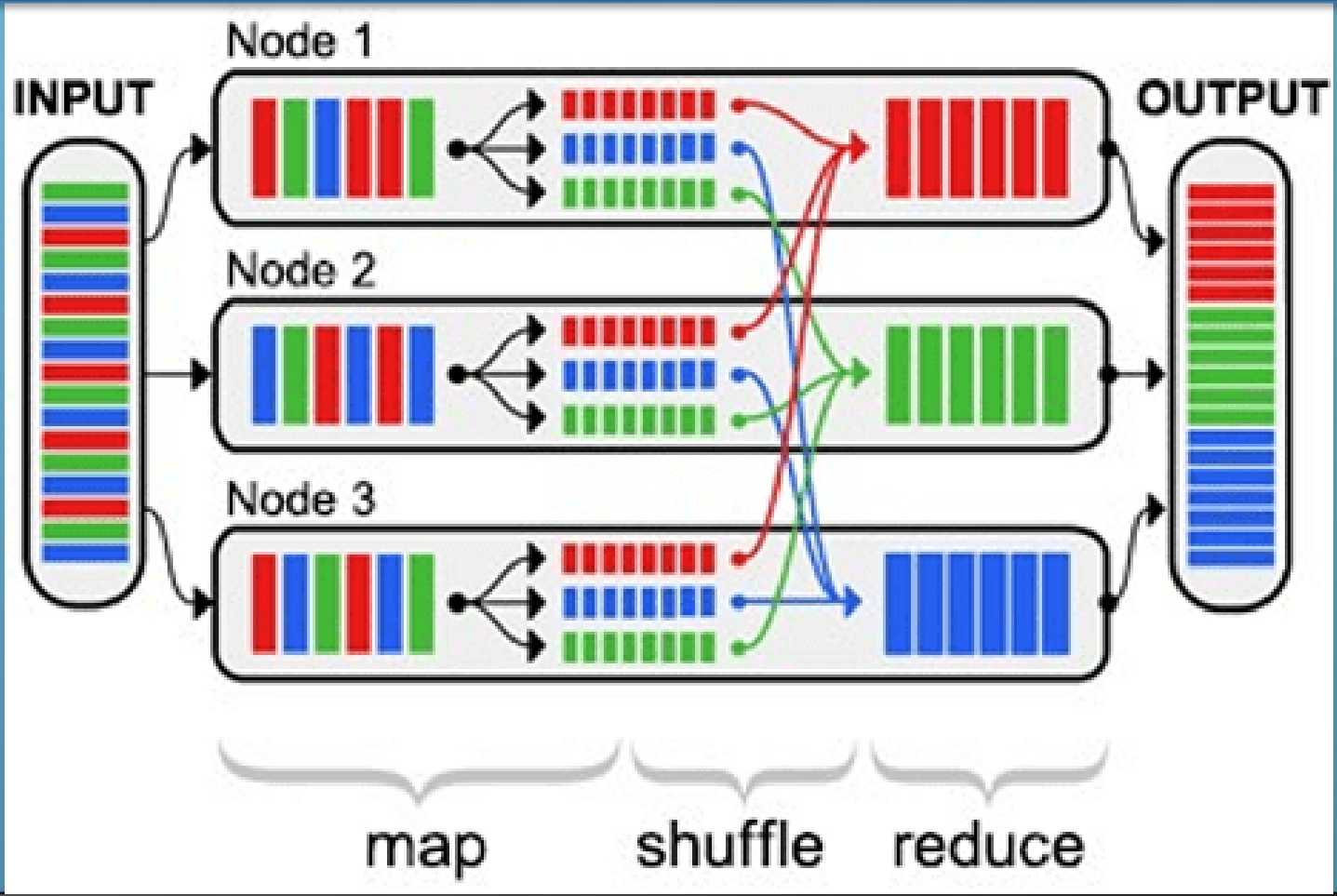


Есть несколько библиотек, которые работают поверх Spark Core, в том числе Spark SQL, который позволяет запускать SQL-подобные команды для распределенных наборов данных, MLLib для машинного обучения, GraphX для проблем с графами и потоковая передача, которая позволяет вводить непрерывную потоковую передачу. данные журнала.

SPARK ИЛИ HADOOP:

	1.	2.	3.	4.	5.	6.
Hadoop	Hadoop — это платформа с открытым исходным кодом, в которой используется алгоритм MapReduce.	Модель Hadoop MapReduce выполняет чтение и запись с диска, что снижает скорость обработки.	Hadoop разработан для эффективной пакетной обработки	Hadoop — это вычислительная среда с высокой задержкой, не имеющая интерактивного режима.	С Hadoop MapReduce разработчик может обрабатывать данные только в пакетном режиме.	Hadoop - более дешевый вариант
Spark	Spark - это высокоскоростная технология кластерных вычислений, которая расширяет модель MapReduce	Spark сокращает количество циклов чтения / записи на диск и сохраняет промежуточные данные в памяти, что увеличивает скорость обработки.	Spark разработан для эффективной обработки данных в реальном времени.	Spark — это вычислитель с низкой задержкой, который может обрабатывать данные в интерактивном режиме.	Spark может обрабатывать данные в реальном времени из таких событий, как твиттер, фейсбук.	Spark требует много оперативной памяти, удорожает кластер.

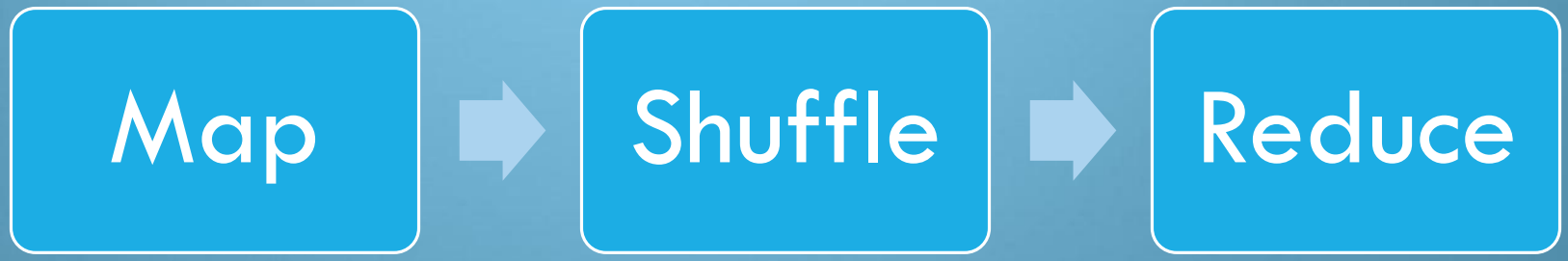
MAP-REDUCE



MapReduce можно по праву назвать главной технологией Big Data, т.к. она изначально ориентирована на параллельные вычисления в распределенных кластерах. Суть MapReduce состоит в разделении информационного массива на части, параллельной обработки каждой части на отдельном узле и финального объединения всех результатов.

ПАТТЕРН MAP-REDUCE

- Относительно универсальная и простая модель для распараллеливания обработки данных.
- Не требует перемещения всех данных на один узел
- Нужно написать только функции map() и reduce()



Каждый узел выполняет функцию map() над своей частью входных данных

Input -> (Key, Value)

Данные перераспределяются по ключам партиционирования так, что все данные с одним значением ключа попадают на один узел

Каждый узел выполняет функцию reduce()
для данных с одним значением ключа

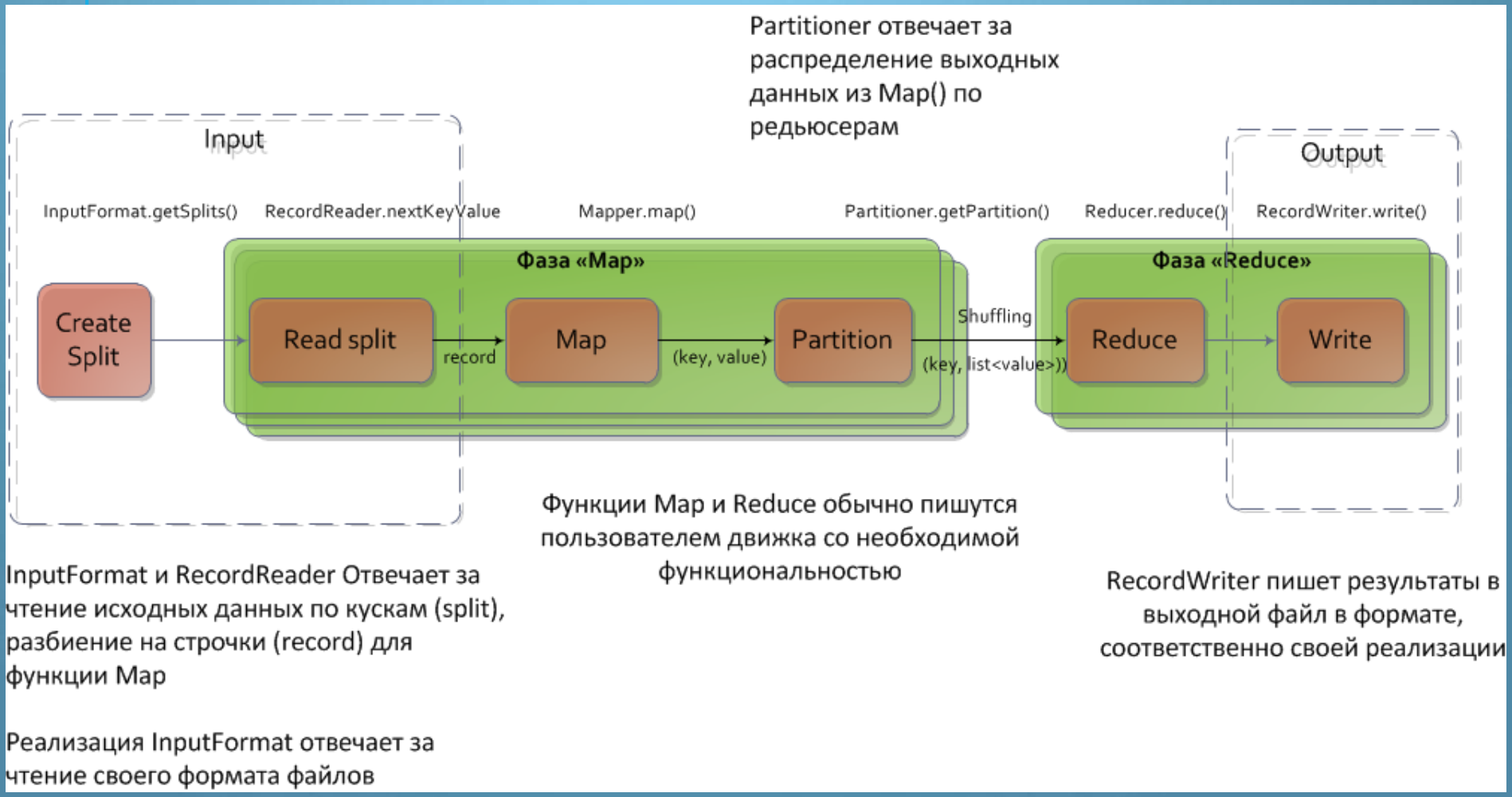
(Key, List<Value>) -> Output

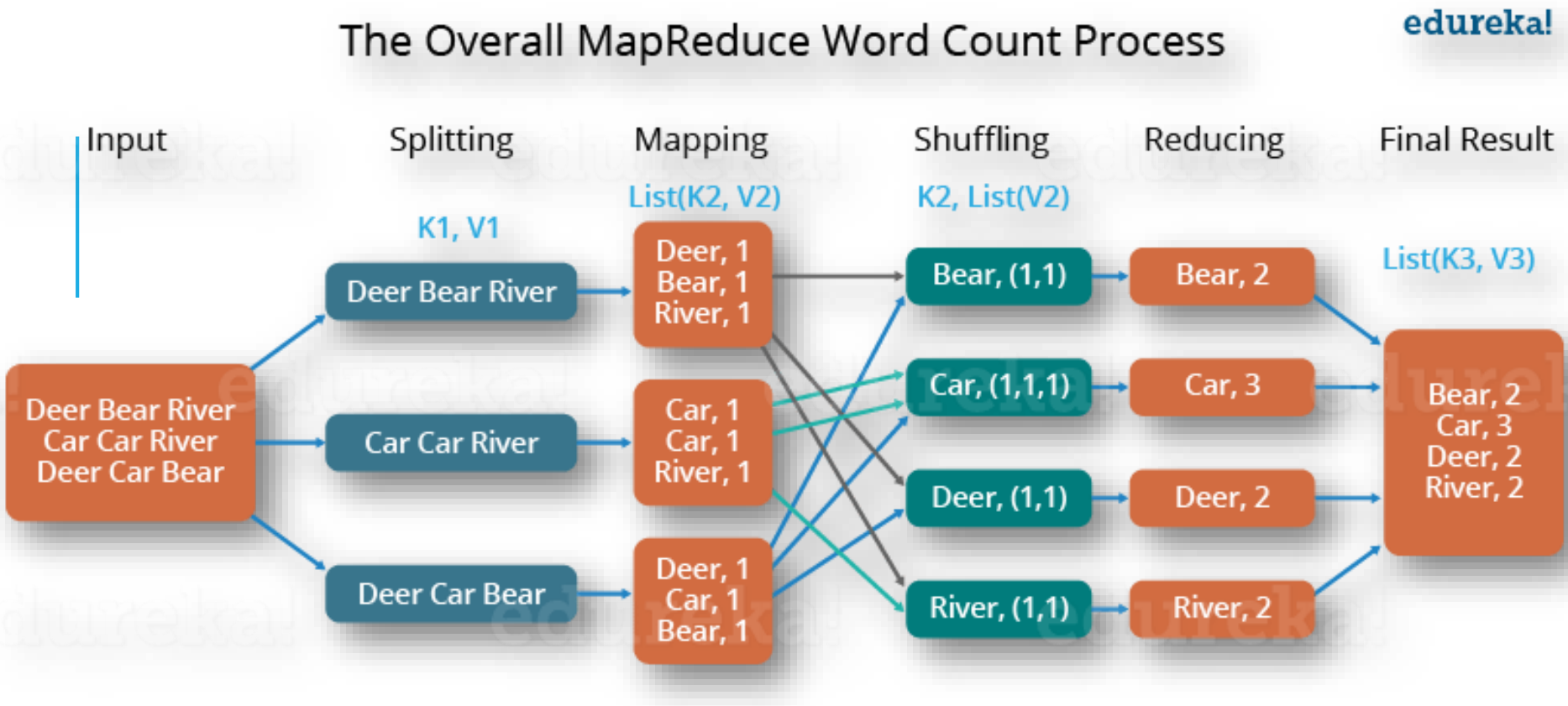
HADOOP MAP-REDUCE

1. Map-задачи состоят из шагов:
   a) *record reader – чтение записи из исходных данных (HDFS блока)*
   b) *mapper – вызов функции map(input) -> (key, value)*
   c) *combiner – (опционально) предварительная агрегация данных*
   d) *partitioner* – вычисление номера reducer-а по ключу
2. Выходом фазы map является набор keys-values, которые сохраняются на диск
3. Reduce-задачи состоят из шагов:
   a) shuffle – чтение «своих» данных с мапперов через HTTP
   b) sort – сортировка по ключу (происходит по мере забора данных в shuffle)
   c) *reduce* – вызов метода reduce для каждого ключа
   d) *output format* – записывает результат на HDFS

- Узлы, на которых выполняются map-задачи обычно соответствуют узлам, где хранится та часть данных, которую они обрабатывают. Таким образом нет необходимости пересылать эти данные по сети (Data Locality)
- Фреймворк старается использовать минимум оперативной памяти и повторно использует Java-объекты, чтобы избежать затрат на сборку мусора (Garbage collection)

HADOOP MAP-REDUCE





Функция

map превращает входной документ в набор пар (слово, 1),

shuffle прозрачно для нас превращает это в пары (слово, [1,1,1,1,1,1]),

reduce суммирует эти единички, возвращая финальный ответ для слова.

```
def map(doc):
    for word in doc:
        yield word, 1

def reduce(word, values):
    yield word, sum(values)
```

ТИПЫ ЗАДАЧ ДЛЯ MAPREDUCE

Агрегация (summarization) данных

- Численная: avg(), count(), sum(), stddev() group by …
- Инвертированный индекс
- Подсчет количества

Фильтрация

- Отсев, Top N, Distinct

Изменение структуры данных

Объединение данных (Join)

Сортировка данных

ЧИСЛЕННАЯ АГРЕГАЦИЯ ДАННЫХ (1/2)

Для каждого пользователя, посчитать количество заданных вопросов, а также минимальный, максимальный и средний рейтинг (“score”) его вопросов

id	owner	score
1	John	-5
2	John	10
3	Alex	24
4	Alex	34
5	John	45

MAP

Mapper 1:

Key	Value			
	count	min	max	sum
John	1	-5	-5	-5
John	1	10	10	10
Alex	1	24	24	24

Mapper 2:

Key	Value			
	count	min	max	sum
Alex	1	34	34	34
John	1	45	45	45

COMBINE

Key	Value			
	count	min	max	sum
John	2	-5	10	-5+10
Alex	1	24	24	24

Key	Value			
	count	min	max	sum
Alex	1	34	34	34
John	1	45	45	45

ЧИСЛЕННАЯ АГРЕГАЦИЯ ДАННЫХ (2/2)

Пересылка по сети

SHUFFLE

REDUCE

Mapper 1:

Key	Value			
	count	min	max	sum
John	2	-5	10	-5+10
Alex	1	24	24	24

Mapper

Key	Value			
	count	min	max	sum
Alex	1	34	34	34
John	1	45	45	45

Reducer 1:

Key	Value			
	count	min	max	sum
Alex	1	24	24	24
Alex	1	34	34	34

Reducer 2:

Key	Value			
	count	min	max	sum
John	2	-5	10	-5+10
John	1	45	45	45

Key	Value			
	count	min	max	sum
Alex	2	24	34	58

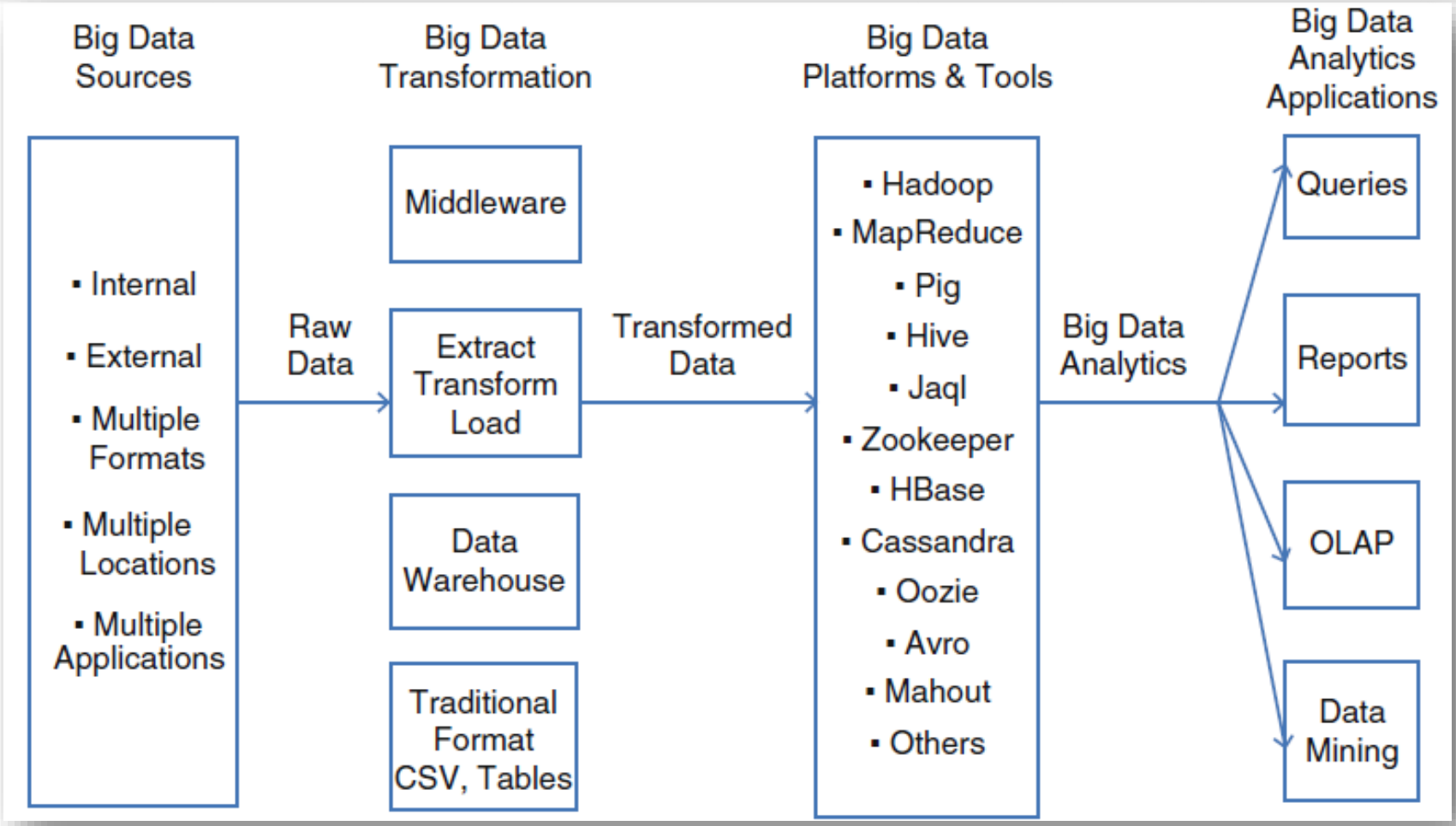
запись в файл /part-r-00000

Key	Value			
	count	min	max	sum
John	3	-5	45	50

запись в файл /part-r-00001

ТАК ЧТО ЖЕ ТАКОЕ БОЛЬШИЕ ДАННЫЕ?

Популярное определение больших данных по Гартнеру: “Большие данные — это информационные активы, которые характеризуются большим объёмом, высокой скоростью и/или многообразием, а также требуют экономически эффективных инновационных форм обработки информации, что приводит к усиленному пониманию, улучшению принятия решений и автоматизации процессов.”



ТАК ЧТО ЖЕ ТАКОЕ БОЛЬШИЕ ДАННЫЕ?

Трудно предоставить решение с ETL для «10 V». Как работать с объемами? Неструктурированными данными? Обеспечивать Скорость? и т.п.

В ЭТОМ МИРЕ НЕ ВРУТ ТОЛЬКО ЦИФРЫ, А ЕСЛИ ВРУТ И ОНИ, ЗНАЧИТ ВЫ НЕ УМЕЕТЕ ИХ ГОТОВИТЬ С)