1.ПОНЯТИЕ БОЛЬШИХ ДАННЫХ. ИСТОРИКО-ФИЛОСОФСКИЕ АСПЕКТЫ

"Мир специально задуман так, чтобы любое познавательное усилие ума становилось новой цепью загадок"

1.1. История феномена

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»

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. Планшеты, электронные книги, аудиокниги, Интернет, Google, Смартфоны, виртуальной реальности, искусственного интеллекта. ***Станислав Лем.*** **из статьи "Информационный барьер?" 1993 год.**

В настоящее время ежегодно **публикуется** 1,5 миллиона **статей**

1.2. Философское прочтение

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

Исходно понятие данных – философское, оно возникает в эпистемологии при рассмотрении основой проблемы гносеологии – познаваемости мира, поиска и осмысления истины. Процедуры верификации или фальсификации данных создают информацию, осмысление истины создает знание.

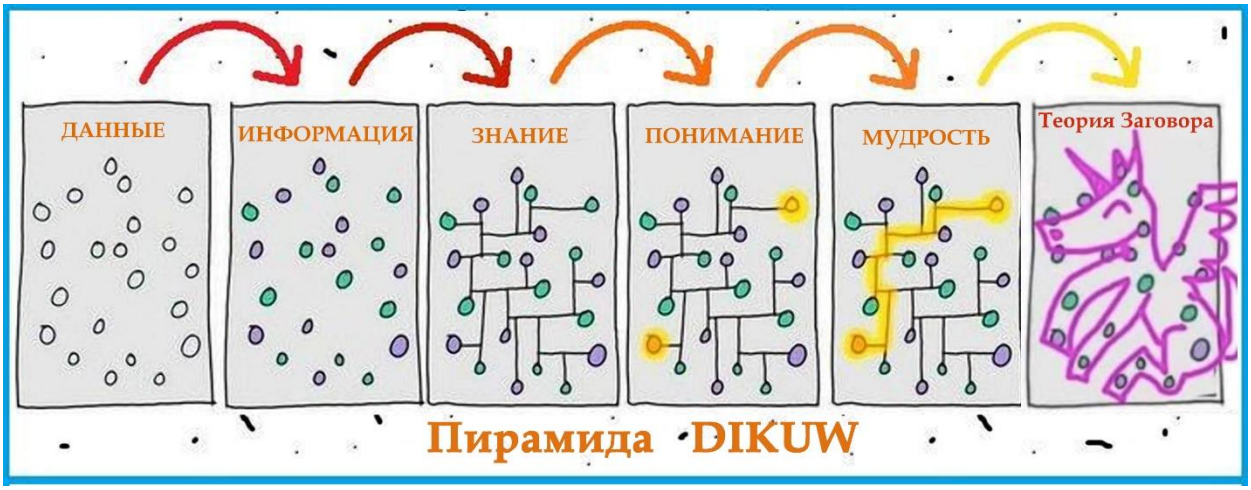

Рис.1 Когнитивная пирамида Data-Information-Knowledge-Understanding-Wisdom (DIKUW)

Точек как будто столько же. Но к ним добавляются метаданные - дополнительные измерения. Мы не можем подходить к оценке этих понятий с линейной и даже численной меркой.

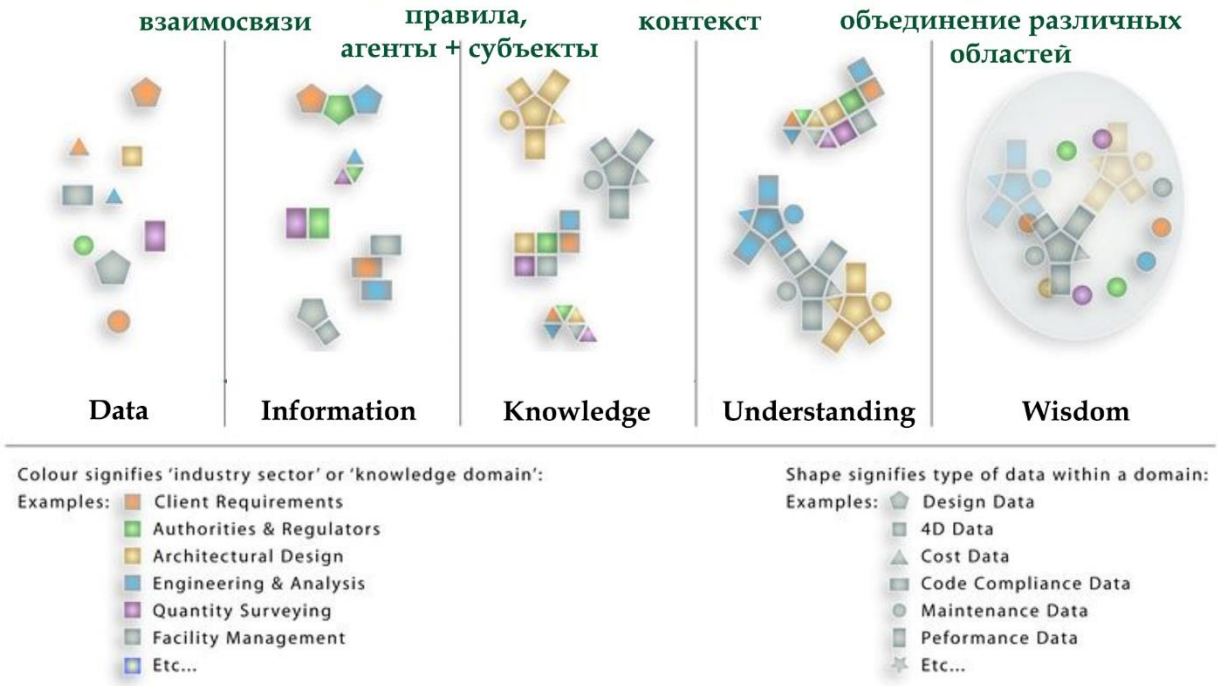

Рис.2 Когнитивная метамодель DIKUW

Философия рассматривает преобразование сведений в данные, данных в информацию, а информации – в знания. Истинность сведений субъективна. Сведения, выраженные в формальном представлении, являются данными.

Обработка данных позволяет определить сколько в них содержится информации. При осмыслении информации экспертом создаются знания.

	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
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.

Таблица.1, часть1. Сводная таблица когнитивной метамодели

	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.

Таблица.1, часть2. Сводная таблица когнитивной метамодели

1.3. Математический подход:

Большие Данные – это наборы данных, которые по мере увеличения количества полей (столбцов) предлагают большую **статистическую мощность**. Т. е. уменьшается вероятность совершить ошибку первого (отвергнуть верную нулевую гипотезу) и второго рода (принять неверную нулевую гипотезу).

Нулевая гипотеза H_0 — некоторое предположение о распределении вероятностей, породившем наблюдаемую выборку данных $x^m = (x_1, \ldots, x_m)$. Статистический тест позволяет проверить, согласуется ли наблюдаемая выборка с

этим распределением (тогда нулевая гипотеза принимается), или не согласуется (тогда нулевая гипотеза отвергается).

Нулевая гипотеза может быть принята или отвергнута только с некоторой вероятностью, см. Уровень значимости

Нулевая гипотеза формулируется таким образом, чтобы при условии её истинности можно было вывести функцию распределения $F(T)$ выбранной статистики критерия $T(x^m)$.

Ошибка, состоящая в принятии нулевой гипотезы, когда она ложна, качественно отличается от ошибки, состоящей в отвержении гипотезы, когда она истинна. Эта разница очень существенна вследствие того, что различна значимость этих ошибок. Проиллюстрируем вышесказанное на следующем примере.[2]

Рассмотрим случай, когда предпринимается действие a_2, в то время, когда a_1 является более предпочтительным. Это означает, что вследствие неточностей в проведении эксперимента партия нетоксичного лекарства классифицировалась как опасная. Последствия ошибки могут выражаться в финансовом убытке и в увеличении стоимости лекарства. Однако случайное отвержение совершенно безопасного лекарства, очевидно, менее нежелательно, чем, пусть даже изредка происходящие гибели пациентов. Отвержение нетоксичной партии лекарства – ошибка второго рода.

Ошибка первого рода («ложная тревога») состоит в том, что гипотеза H_0 будет отвергнута, хотя на самом деле она правильная. Вероятность допустить такую ошибку называют ***уровнем значимости*** и обозначают буквой α («альфа»). **Допустимая вероятность ошибки первого рода** *(Ркр)* может быть равна 5% или 1% (0.05 или 0.01).

Ошибка второго рода («пропуск цели») состоит в том, что гипотеза H_0 будет принята, но на самом деле она неправильная. Вероятность совершить эту ошибку обозначают буквой β («бета»). Значение $1-\beta$ называют ***мощностью критерия*** – это вероятность отвержения неправильной гипотезы.

2. МЕСТОРАСПОЛОЖЕНИЕ БОЛЬШИХ ДАННЫХ В МИРОВОЙ ЭКОНОМИКЕ.

2.1. Основные определения и понятия

Хинчклиф делит подходы к Big Data на три группы: Быстрые Данные (Fast Data), их объем измеряется терабайтами; Большая Аналитика (Big Analytics) —

петабайтные данные и Глубокое Проникновение (Deep Insight) — экзабайты, зеттабайты. Группы различаются между собой не только оперируемыми объемами данных, но и качеством решения по их обработки.

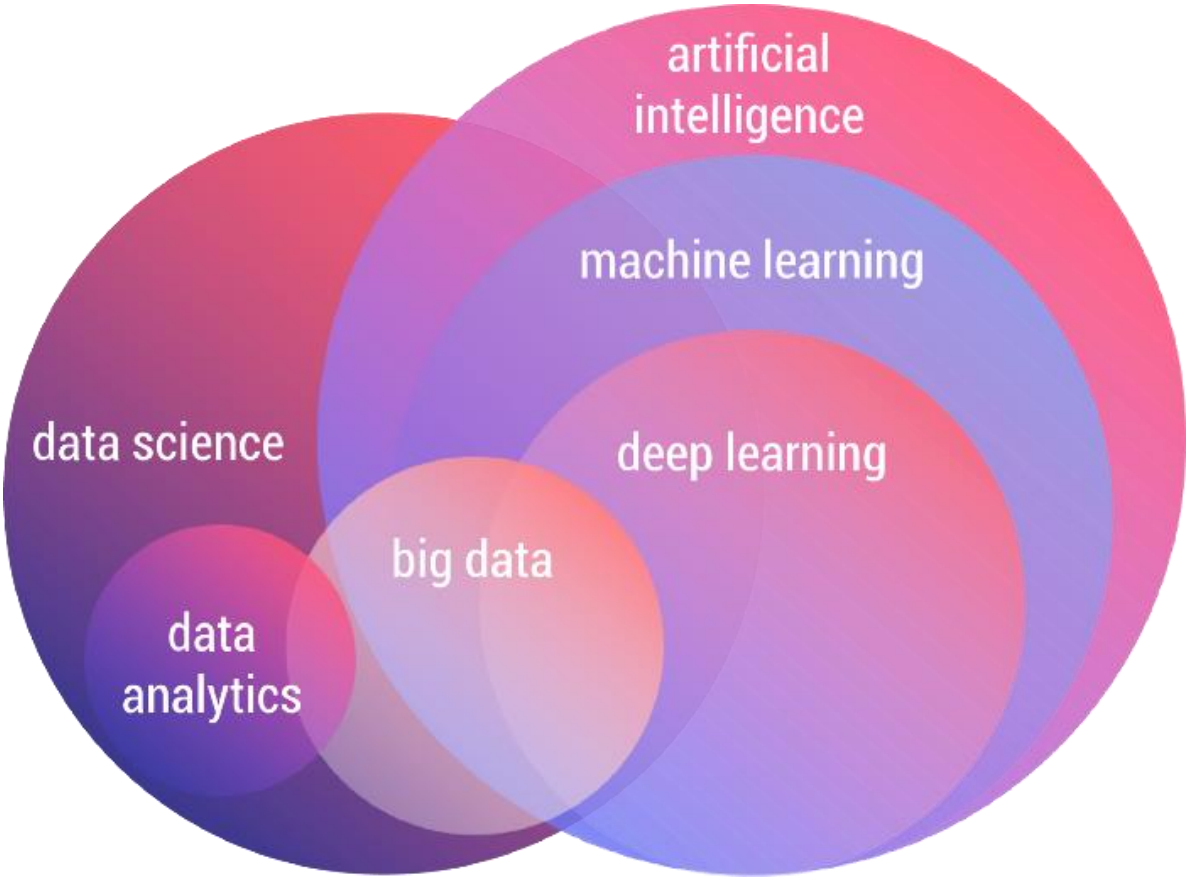

Рисунок 3. Таксономия искусственного интеллекта и науки о данных

Термин большие данные не имеют строгого определения. Нельзя провести четкую границу — это 10 терабайт или 10 мегабайт? Слово «большое» — это как «один, два, много» у первобытных племен. Вообще говоря, большие данные как задача возникла, когда появились места для их хранения и сети для их передачи. Большого адронного коллайдера, IoT, Youtube

Три группы Big Data по Дайон Хинчклиф:

- 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

- Глубокое Проникновение (Deep Insight) — экзабайты, зеттабайты; возможно обнаружение знаний и закономерностей, априорно неизвестных.

(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 »

Мотивация	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

Таблица 3. Основные задачи Больших Данных в мировой экономике.

Рисунок 4. Сферы применения Больших Данных в мировой экономике.

В сущности, понятие больших данных подразумевает работу с информацией огромного объема и разнообразного состава, весьма часто обновляемой и находящейся в разных источниках в целях увеличения эффективности работы, создания новых продуктов и повышения конкурентоспособности. Консалтинговая компания Forrester дает краткую формулировку: `Большие данные объединяют техники и технологии, которые извлекают смысл из данных на экстремальном пределе практичности`.

- Большие данные предназначены для обработки более значительных объемов информации, чем бизнес-аналитика, и это, конечно, соответствует традиционному определению больших данных.
- Большие данные предназначены для обработки более оперативно получаемых и меняющихся сведений, что означает глубокое их исследование и интерактивность. В некоторых случаях результаты формируются быстрее, чем загружается веб-страница.
- Большие данные предназначены для оперирования неструктурированными данными, способы использования которых мы начинаем изучать только после того, как смогли наладить их сбор и хранение, и нам требуются алгоритмы и возможность диалога для облегчения поиска тенденций, содержащихся внутри этих массивов.

Ниже показано несколько реальных сценариев, которые дают нам гораздо лучшее понимание четырех V и определения больших данных:

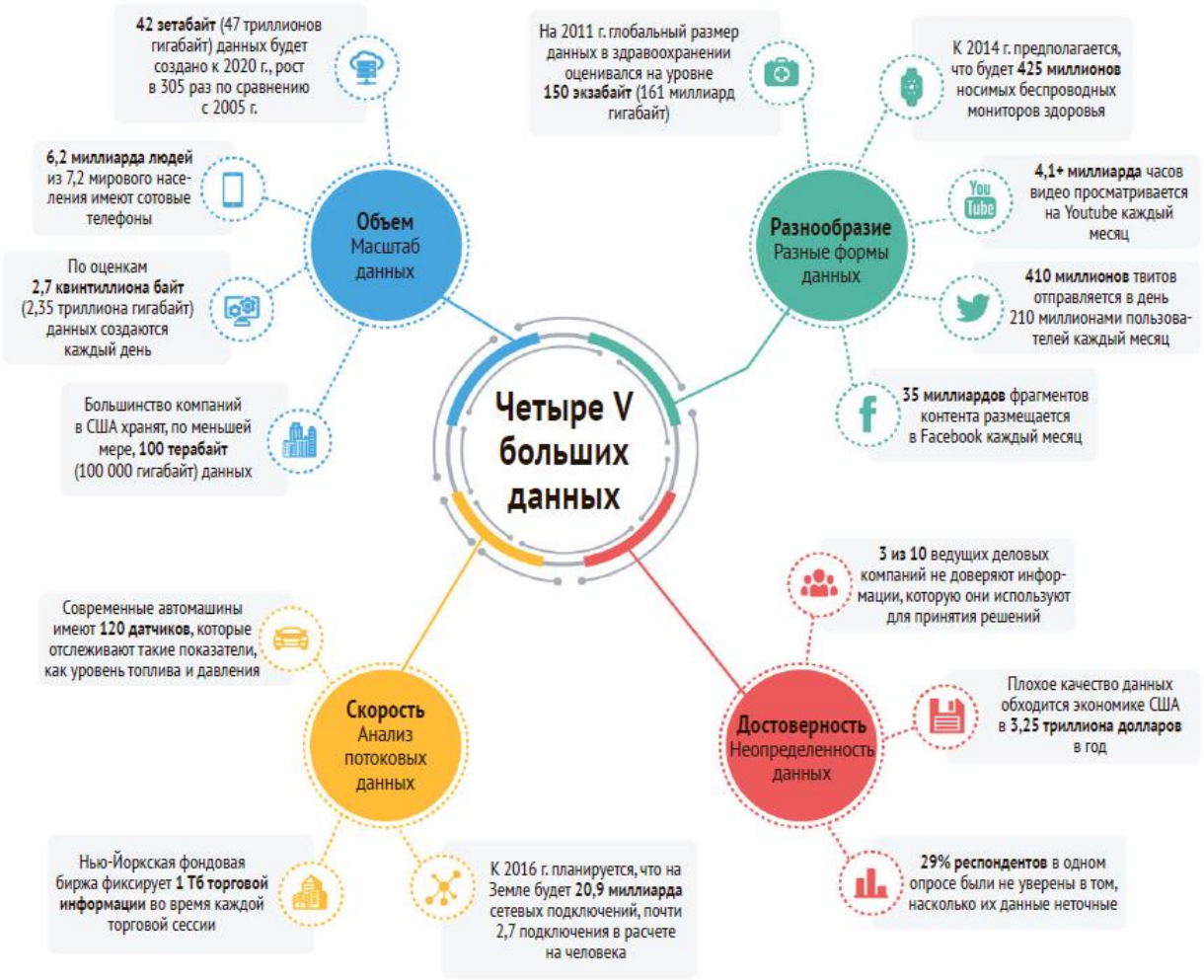

Рисунок 5. Четыре больших V Больших данных

2.2. Текущее состояние. Самые важные цифры из глобального отчета Digital 2020

Значение цифровых технологий в нашей жизни достигло новых высот, и все больше людей проводят все больше времени в интернете, решая там все больше задач:

- Количество интернет-пользователей в мире выросло до 4,54 миллиарда, что на 7% больше прошлогоднего значения (+ 298 миллионов новых пользователей в сравнении с данными на январь 2019 года).
- В январе 2020 года в мире насчитывалось 3,80 миллиарда пользователей социальных сетей, аудитория соцмедиа выросла на 9% по сравнению с 2019 годом (это 321 миллион новых пользователей за год).
- Сегодня более 5,19 миллиарда человек пользуются мобильными телефонами — прирост на 124 миллиона (2,4%) за последний год.

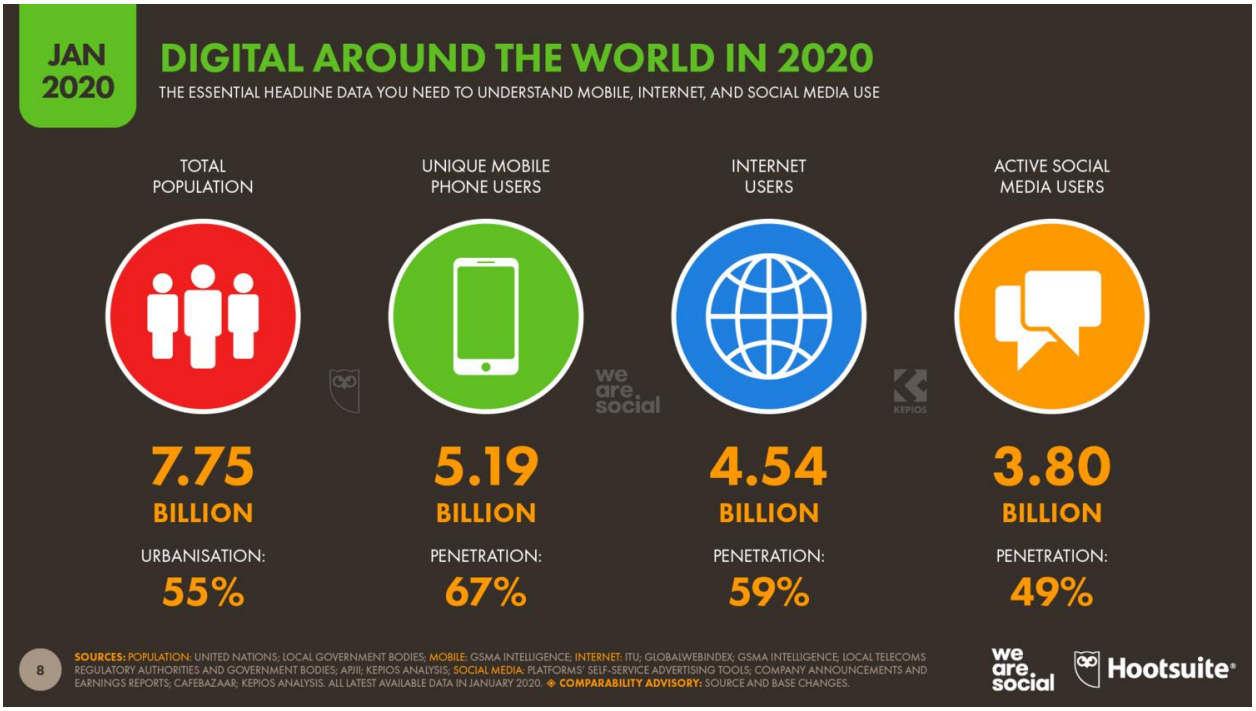

Рисунок 6. Сводная статистика состояния Интернет в мире 2020

Отправлено 473400 твитов, 4500000 видео просматривается на YouTube, 3788140 поисковых запросов выполняется в Google, отправляется 12986111 текстовых сообщений, 1111 пакетов отправляются Amazon, 750 000 песен транслируются в потоковом режиме на Spotify, 49380 фотографий размещаются в Instagram.

Рисунок 7. Визуальная иллюстрация количества данных, создаваемых за минуту (согласно Domo, апрель 2019 г.)

Их источниками могут быть:

- 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.).

3. ДАННЫЕ С ТОЧКИ ЗРЕНИЯ БИЗНЕСА. ЦИФРОВАЯ ТРАНСФОРМАЦИЯ, ИНДУСТРИЯ 4.0.

Data is the new oil.” — Clive Humby

«Управление данными — это процесс приема, хранения, организации и обслуживания данных, созданных и собранных организацией»

Данные в первую очередь рассматриваются как корпоративный актив, используемый для принятия обоснованных бизнес-решений (для получения лучшей отдачи от инвестиций (ROI)) на базе предиктивного, статистического и пространственного анализа, а также для оптимизации и рационализации бизнес-процессов; тем самым снижая затраты и увеличивая выпуск продукции.

Данные — это пресловутая новая нефть и источник жизненной силы Индустрии 4.0.

Рисунок 8. Этапы технической революции.

Как и любой другой бизнес-актив, данные следует защищать, хранить и надлежащим образом оценивать. Часто предприятиям не удается понять все преимущества правильного управления данными, в результате чего они теряют свое конкурентное преимущество.

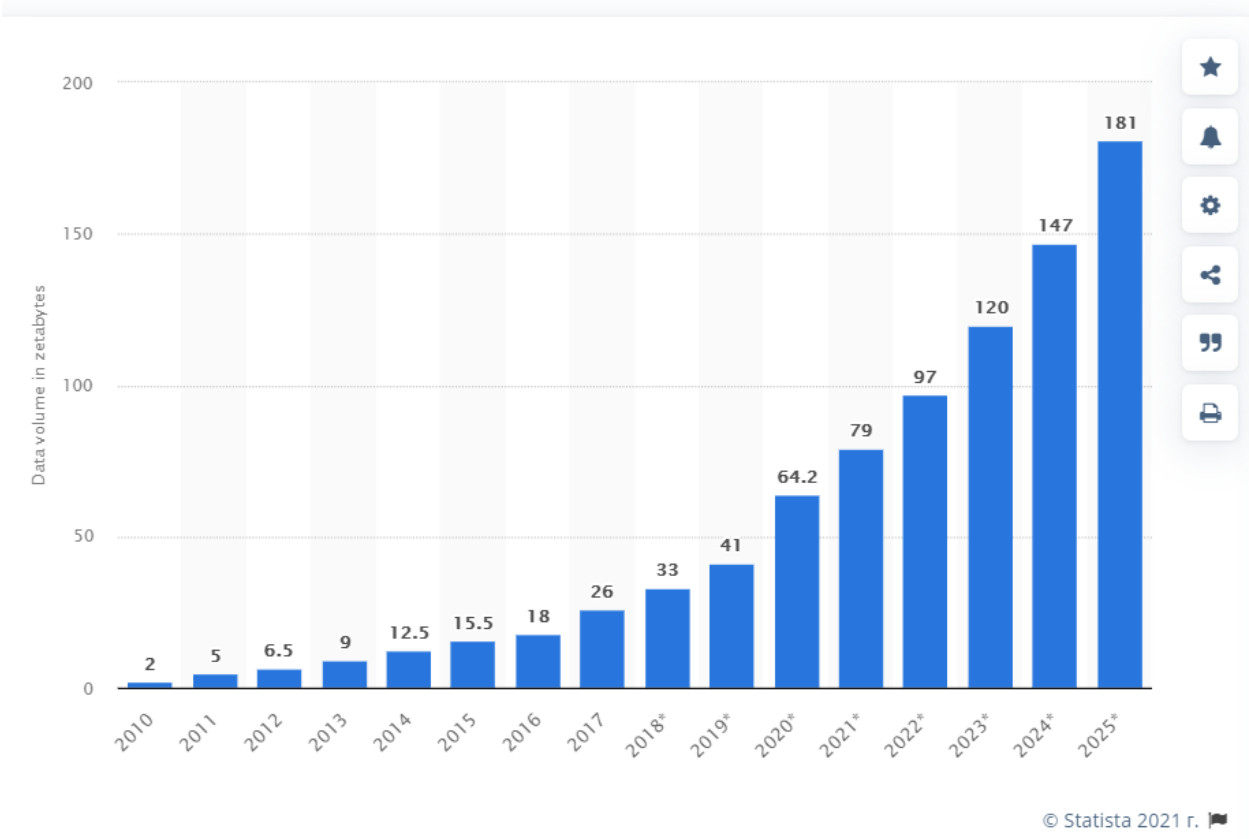

Рисунок 9. Суммарный объем данных Интернета в зетафлопс.

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

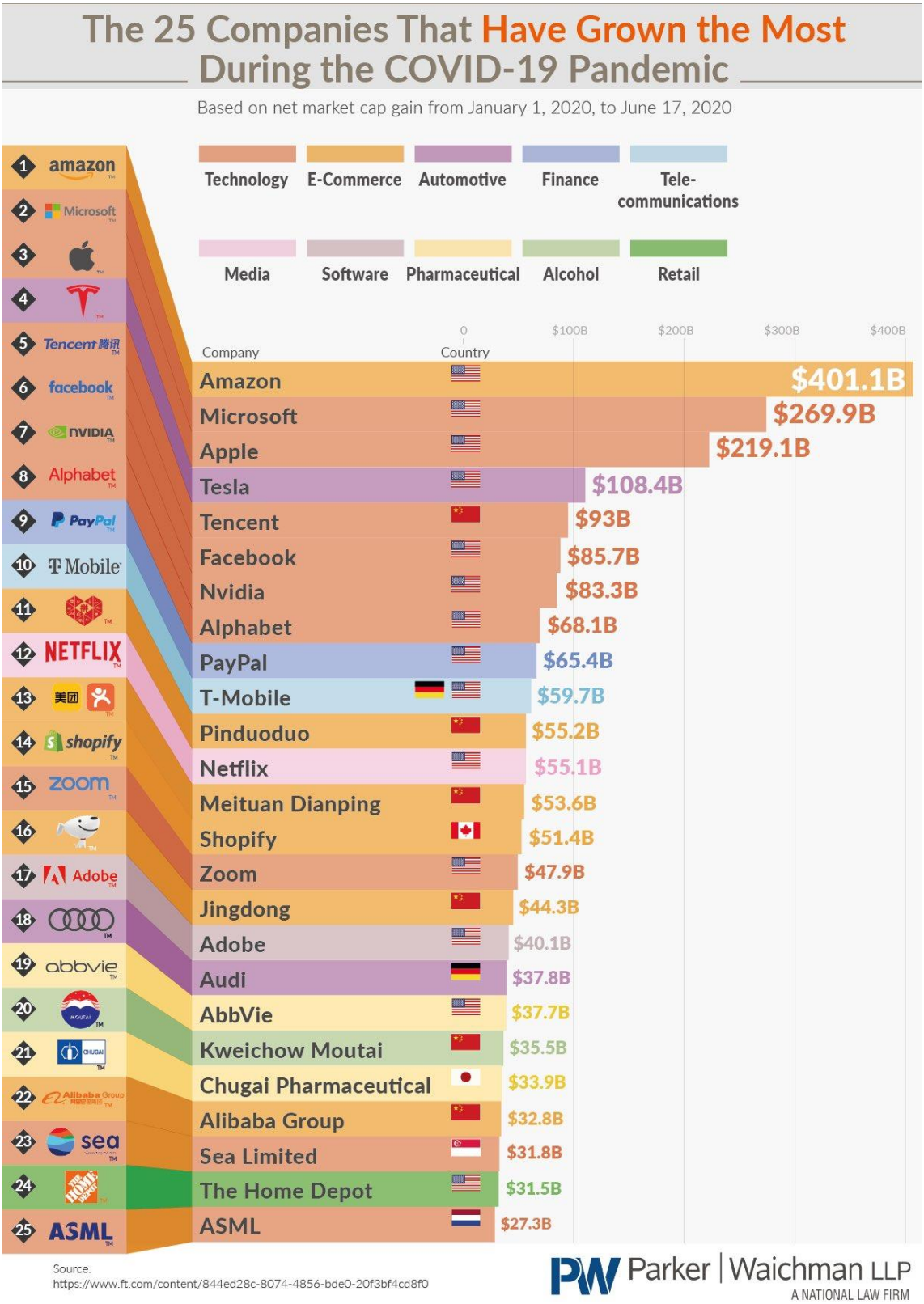

Рисунок 10. Хроника цифровой трансформации

Что же позволяет достичь компаниям такого уровня?

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

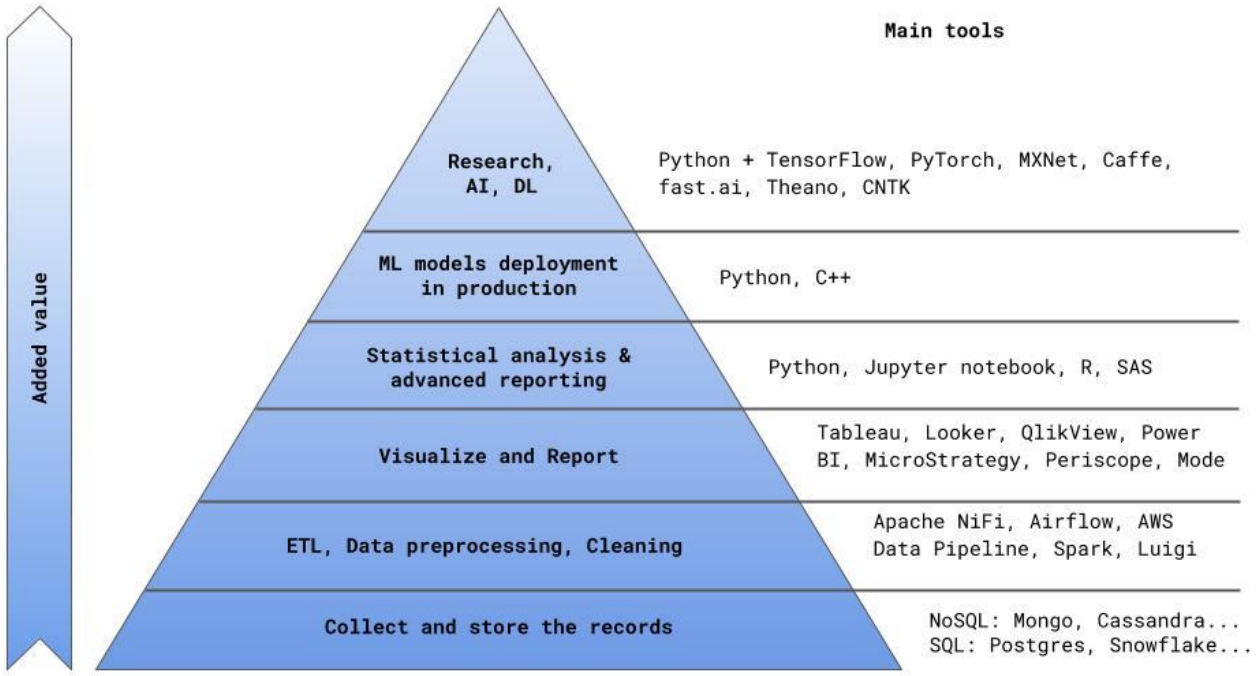

Рисунок 11. Анализ больших данных главная задача в IT

Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — **системы поддержки принятия решений**.

Принять любое управленческое решение, невозможно не обладая необходимой для этого информацией, обычно количественной. Для этого необходимо создание хранилищ данных (Data warehouses), то есть процесс сбора, отсеивания и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа (а нередко и создания аналитических отчетов).

В числе востребованных сегодня методов углубленного анализа:

- Предиктивная аналитика (например, предсказание временных рядов)

- Дескриптивная аналитика (например, факторный анализ)
- Поиск инсайтов (помогает аналитику быстрее доставать интересную информацию)
- Разговорная аналитика (работа с естественным языком)
- Предписывающая аналитика (помогает улучшить качество принятия решений на основе рекомендаций)

5.ПРОГРАММНАЯ АРХИТЕКТУРА БОЛЬШИХ ДАННЫХ

5.1. Основные типы и провайдеры СУБД. Relational DBMS

380 systems in ranking, October 2021

Rank			DBMS	Database Model	Score		
Oct 2021	Sep 2021	Oct 2020			Oct 2021	Sep 2021	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

Рисунок 12. Рейтинг 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.

Схемы таблиц генерируются нормализацией в процессе моделирования данных.

Над отношениями определены некоторые базовые операции:

- классические операции над множеством (объединение, пересечение и разность)
- Выбор (выбор подмножества записей в соответствии с определенными критериями фильтрации для значений атрибутов)
- Проекция (выбор подмножества атрибутов / столбцов таблицы)
- Объединение: специальное соединение нескольких таблиц как комбинация декартова произведения с выделением и проекцией.

Эти базовые операции, а также операции по созданию, изменению и удалению схем таблиц, операции по контролю транзакций и управлению пользователями выполняются с помощью языков баз данных, причем SQL является общепризнанным стандартом для таких языков.

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

За прошедшие годы многие СУБД были расширены нереляционными концепциями, такими как определяемые пользователем типы данных, а не атомарные атрибуты, наследование и иерархии, поэтому их иногда называют объектно-реляционными СУБД.

5.2 Нереляционные СУБД (NOSQL)

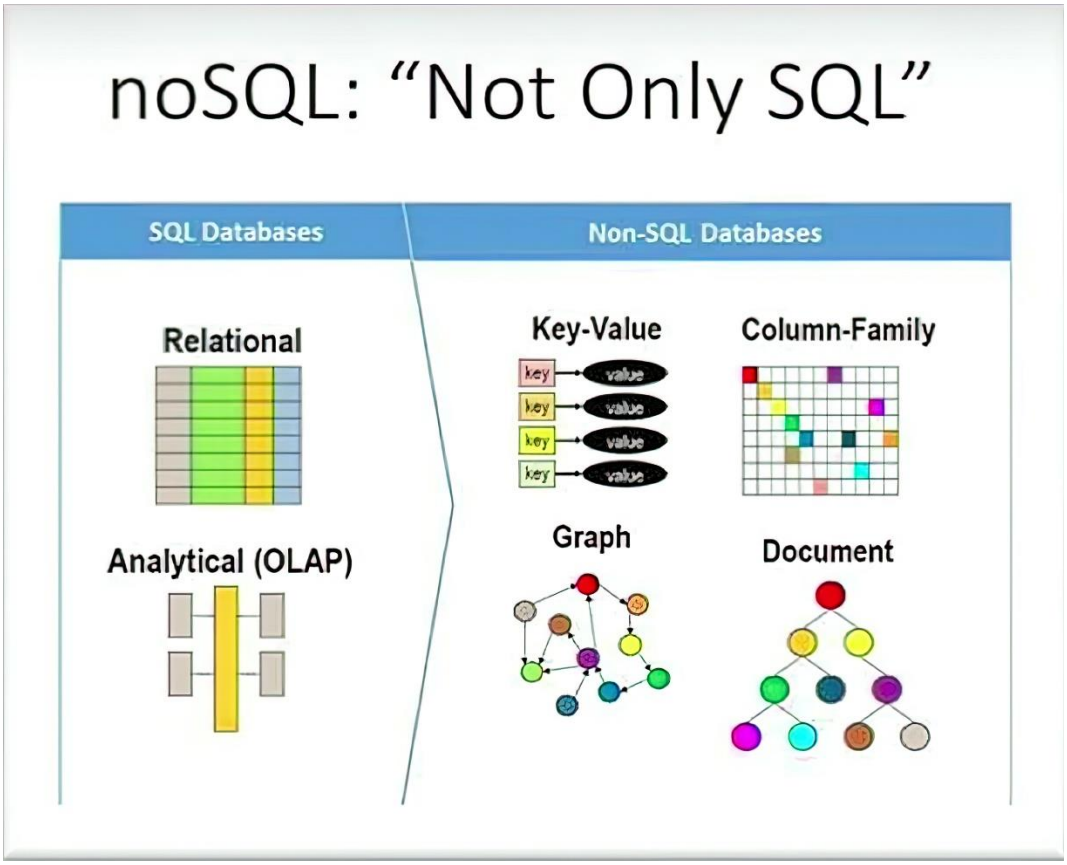

Рисунок 13. Семейство нереляционных СУБД NoSQL

Key-value Stores (это простейший NoSQL)

- Redis
- Amazon DynamoDB
- Microsoft Azure Cosmos DB
- Memcached
- etcd
- Riak

- Хранилища ключей и значений — это наиболее простые типы баз данных NoSQL.
- Создан для обработки огромных объемов данных.
- На основе статьи Amazon Dynamo.
- 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 «ключ-значение» следуют аспектам «доступности» и «разделения» теоремы CAP.
- 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.

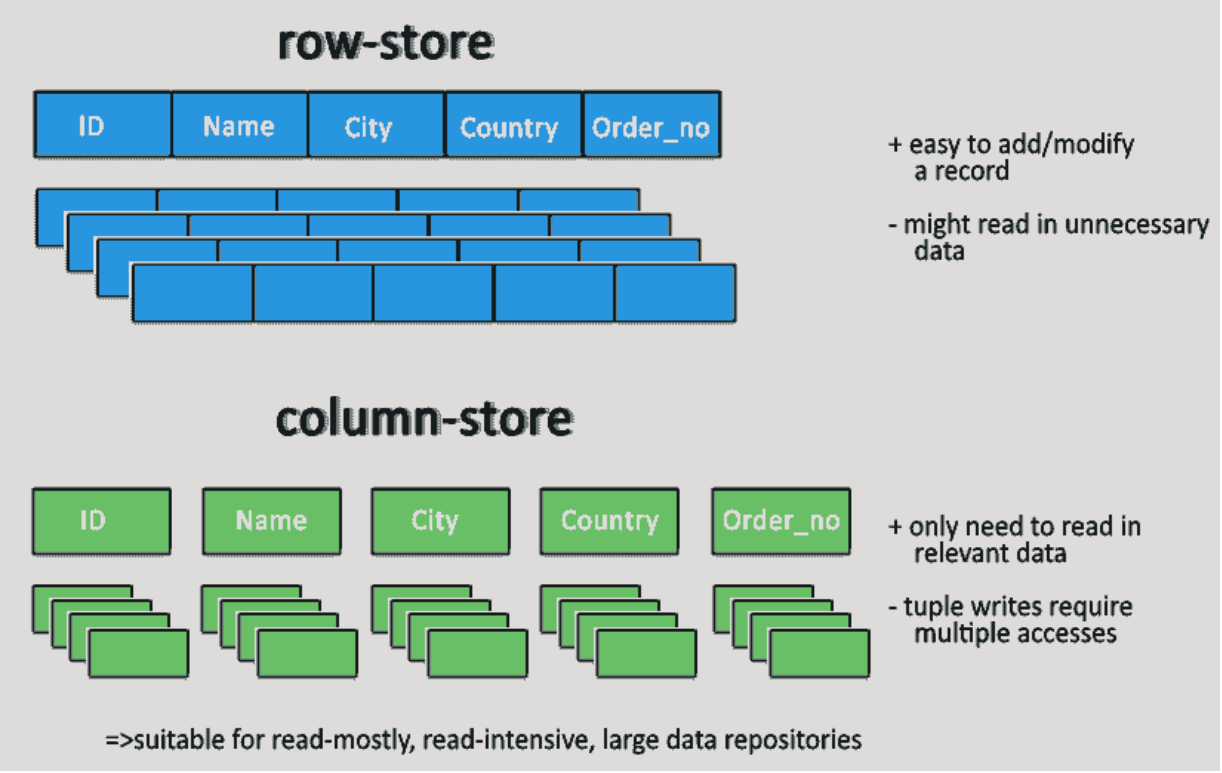

Рисунок 14. Панельное и колоночное представление данных

Хранилища "ключ-значение", вероятно, являются самой простой формой систем управления базами данных . Они могут хранить только пары ключей и значений, а также извлекать значения, когда ключ известен.

Эти простые системы обычно не подходят для сложных приложений. С другой стороны, именно эта простота делает такие системы привлекательными в определенных обстоятельствах. Например, ресурсоэффективные хранилища ключей и значений часто применяются во встроенных системах или как высокопроизводительные внутрипроцессные базы данных.

Рисунок 15. Архитектура хранилищ Ключ-Значение

5.4. ХРАНИЛИЩА ШИРОКИХ СТОЛБЦОВ (ЭТО ОДИН ИЗ ТИПОВ NOSQL)

Расширенная форма хранилищ ключей и значений позволяет сортировать ключи и, таким образом, позволяет выполнять запросы диапазона, а также упорядоченную обработку ключей.

Многие системы предоставляют дополнительные расширения, так что мы видим довольно плавный переход к хранилищам документов и хранилищам с широкими столбцами .

- 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

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 .

Широкие хранилища столбцов разделяют свойство отсутствия схемы с хранилищами документов, однако реализация сильно отличается.

Хранилища с широкими столбцами не следует путать с хранилищами, ориентированными на столбцы, в некоторых реляционных системах . Это внутренняя концепция повышения производительности СУБД для рабочих нагрузок OLAP, в которой данные таблицы хранятся не запись за записью, а столбец за столбцом.

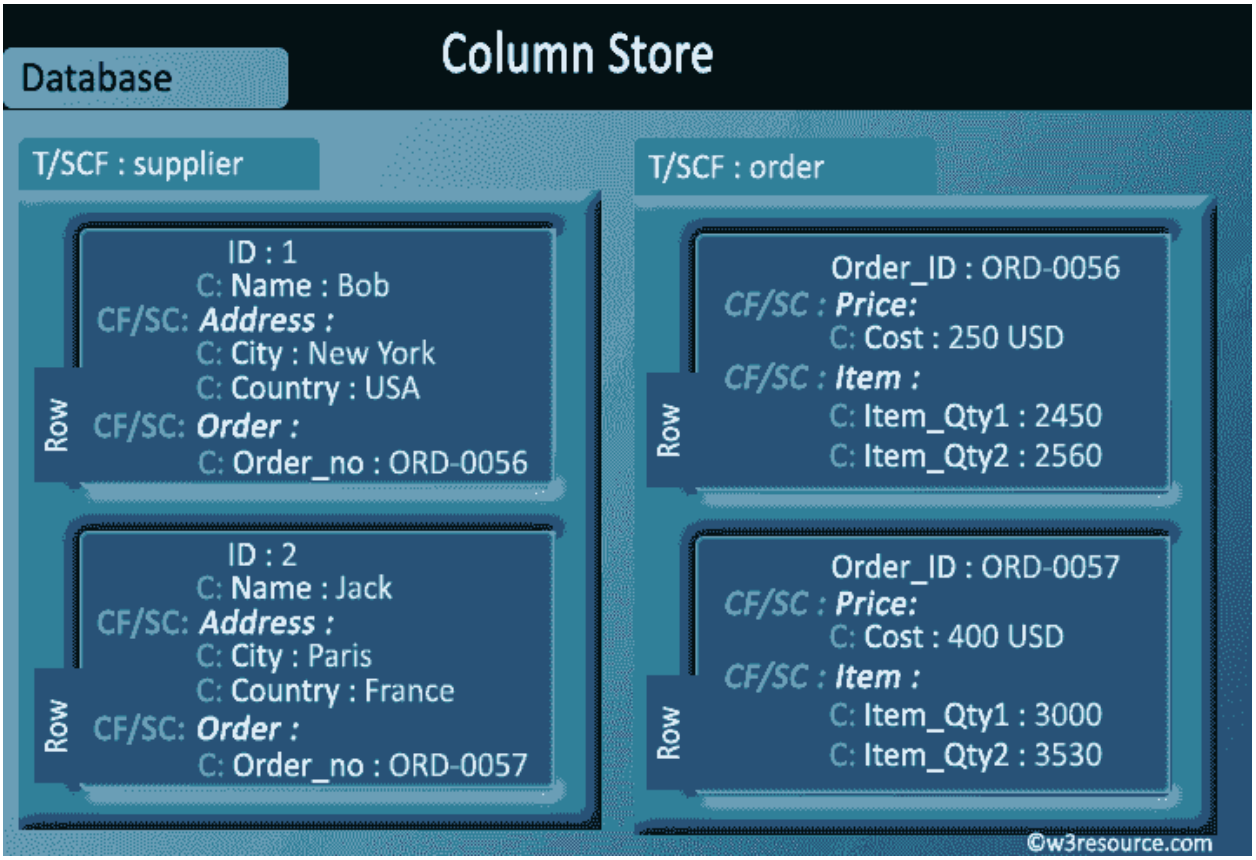

Рисунок 16. Архитектура колоночных хранилищ.

5.5. Document Stores Документно-ориентированные 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, чтобы сгруппировать различные типы данных.
- Документы могут содержать много разных пар ключ-значение, пар ключ-массив или даже вложенных документов.

Вот сравнение классической реляционной модели и модели документа:

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

Таблица 4. Реляционная и документная модели

Хранилища документов, также называемые системами баз данных, ориентированными на документы, характеризуются структурой данных без схем. Это означает:

- Записи не обязательно должны иметь единую структуру, т. Е. Разные записи могут иметь разные столбцы.
- Типы значений отдельных столбцов могут быть разными для каждой записи.
- Столбцы могут иметь более одного значения (массивов).
- Записи могут иметь вложенную структуру.

В хранилищах документов часто используются внутренние обозначения, которые могут обрабатываться непосредственно в приложениях, в основном JSON. Документы JSON, в свою очередь могут храниться в виде чистого текста в хранилищах «ключ-значение» или в системах реляционных баз данных . Однако для этого потребуется обработка структур на стороне клиента, а недостаток заключается в том, что функции, предлагаемые хранилищами документов (например, вторичные индексы), недоступны.

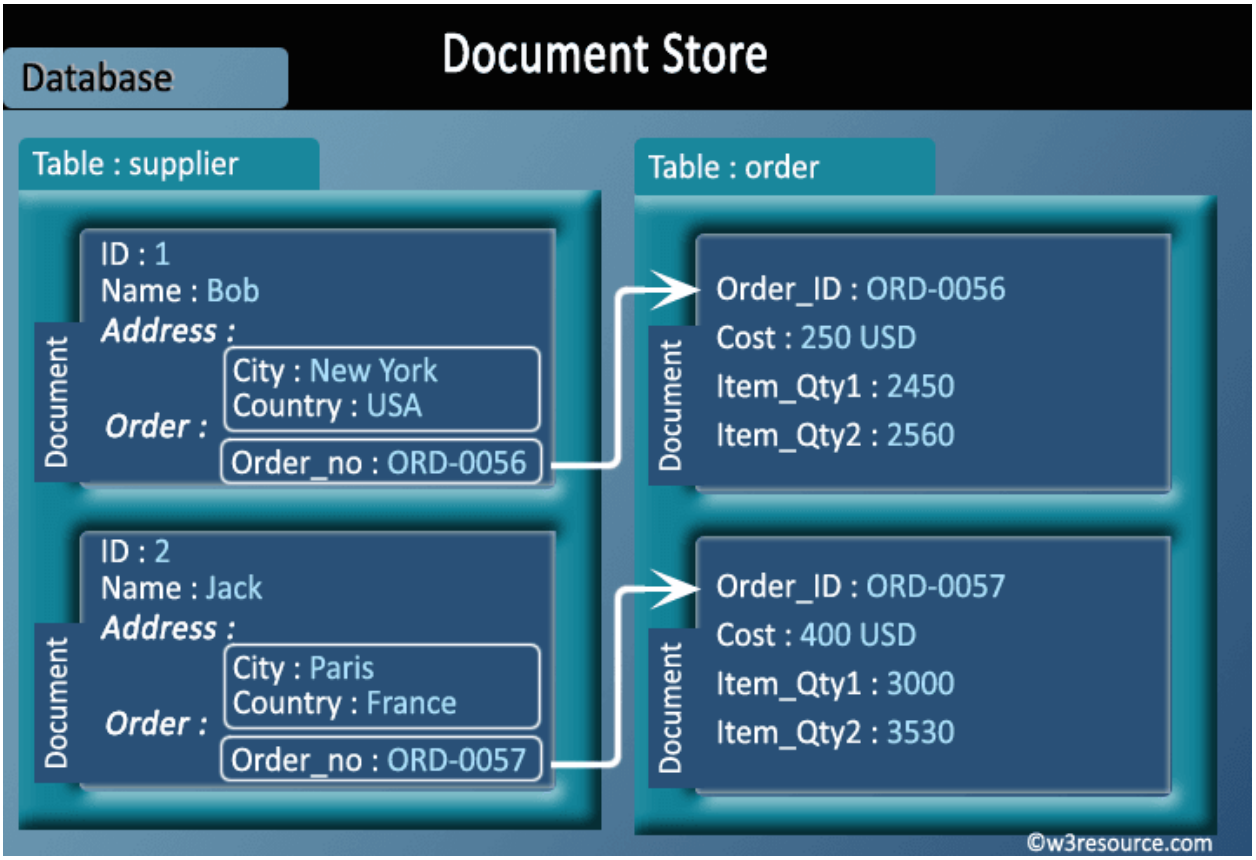

Рисунок 17. Пример реализации документной базы данных Neo4j, ArangoDB, OrientDB, Amazon Neptune

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

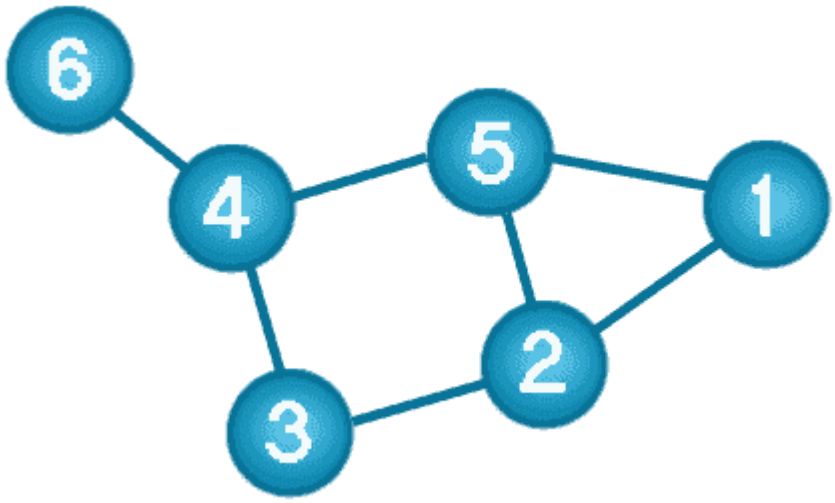

Рисунок 18. Помеченный граф из 6 вершин и 7 ребер.

Что такое графические базы данных?

- Хранит данные в виде графа, набор узлов и ребер
- Он способен 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.
- Индекс для поиска.

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

Таблица 5. Сравнение классической реляционной модели и графовой модели:

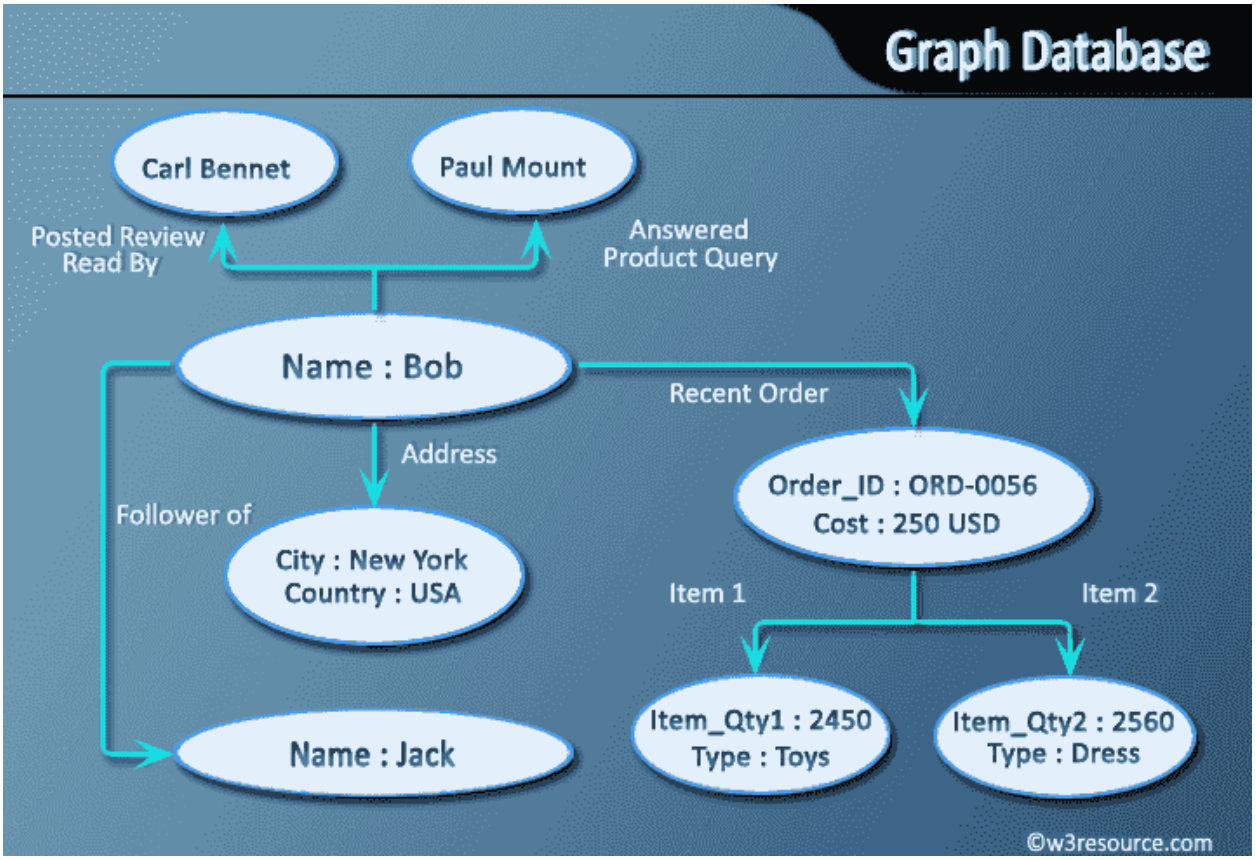

Рисунок 19. Пример реализации графовой БД

6. OLTP И OLAP-ТЕХНОЛОГИИ

6.1: Многомерное представление данных. Общая схема организации хранилища данных. Характеристики, типы и основные отличия технологий OLAP и OLTP. Схемы звезда и снежинка. Агрегирование

Возможности OLAP. MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP), HOLAP (Hybrid OLAP)

В бизнесе ежедневно приходится принимать множество решений: производственные, маркетинговые и кадровые решения; решения, затрагивающие цены, продажи, скидки. Принятие эффективных решений должно происходить на всех уровнях управления организацией, что в конечном итоге приводит к успеху всей организации в целом.

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.

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.

Данная проблема эффективно решается с помощью информационно-аналитических систем, построенных на базе OLAP-технологий (другие названия: OLAP-система, Система бизнес-аналитики, Business Intelligence). OLAP-системы интегрируют существующие системы учёта, предоставляя пользователю инструменты для анализа больших объёмов данных в реальном времени, динамического конструирования отчетов, мониторинга и прогнозирования ключевых бизнес-показателей.

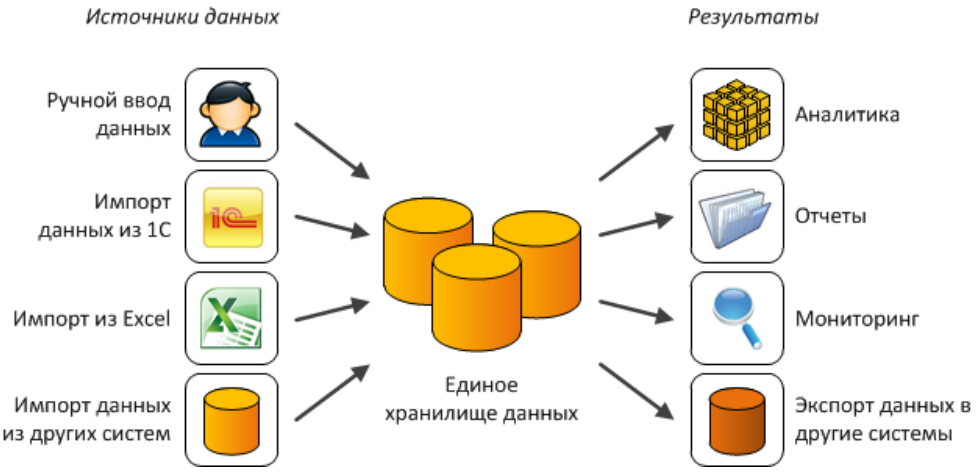

Рисунок 20. Организационная структура единого хранилища данных.

Ключевую роль в управлении компанией играет информация. Как правило, даже небольшие компании используют несколько информационных систем для автоматизации различных сфер деятельности. Получение аналитической отчётности в информационных системах, основанных на традиционных базах данных сопряжено с рядом ограничений:

6.2. Разработка отчётов.

Отчёты формируются очень медленно (зачастую несколько часов), замедляя при этом работу всей информационной системы. Данные, получаемые от различных структурных элементов компании не унифицированы и часто противоречивы.

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. ETL: сила, объединяющая OLTP и OLAP

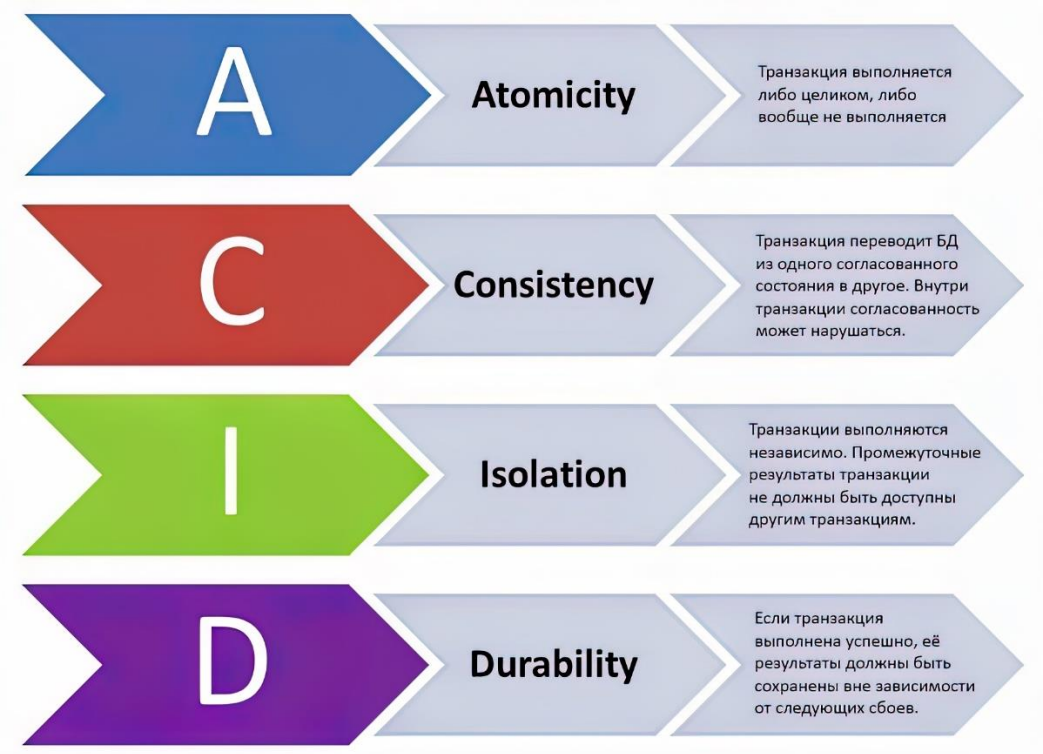

Рисунок 23. Парадигма ACID

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.

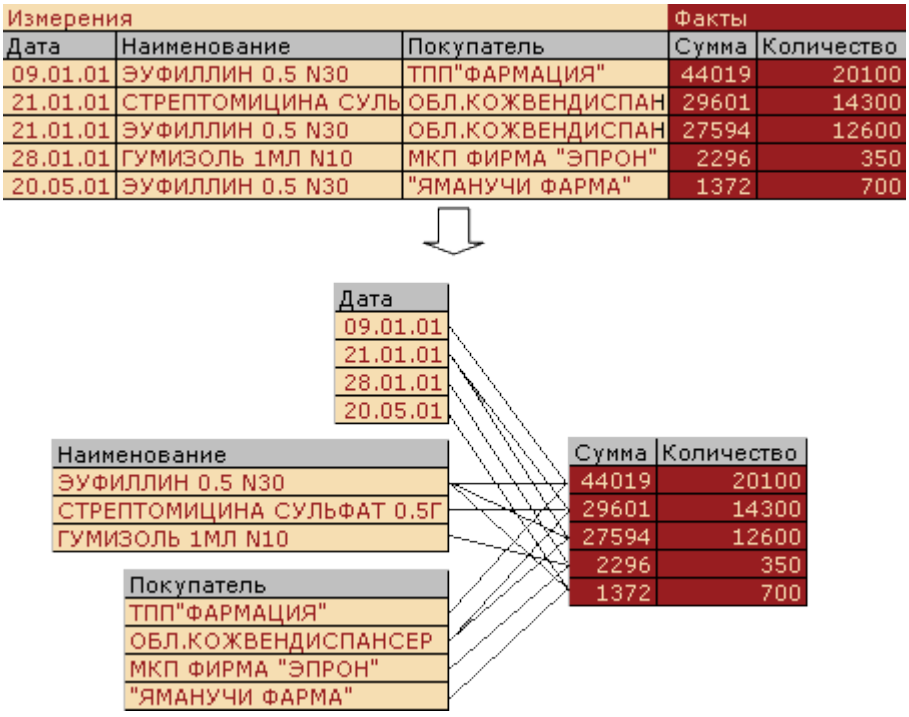

Рисунок 24. OLAP Преобразование схемы базы данных.

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.

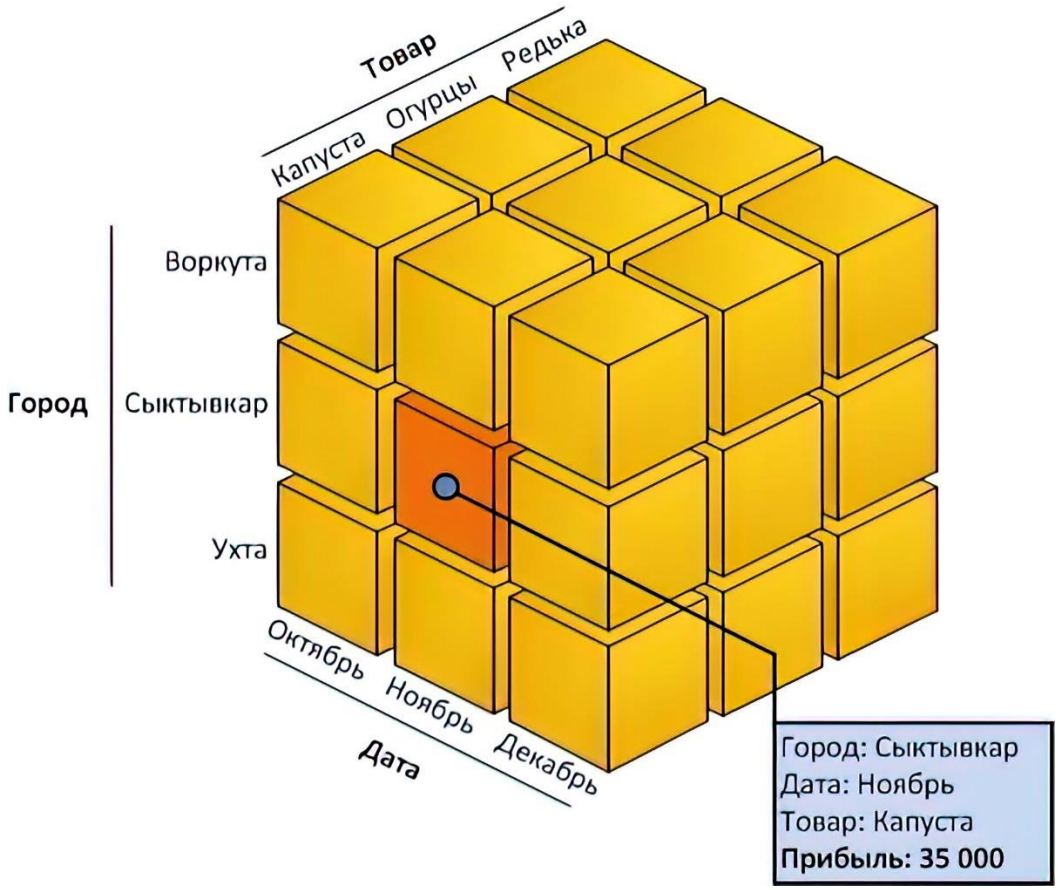

Рисунок 25. Пример OLAP куба.

Создание OLAP-системы на предприятии позволяет:

- 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.

OLAP расшифровывается как Online Analytical Processing Server. Это программная технология, которая позволяет пользователям одновременно анализировать информацию из нескольких систем баз данных. Он основан на многомерной модели данных и позволяет пользователю запрашивать многомерные данные (например, Дели -> 2018 -> Данные о продажах). Базы данных OLAP разделены на один или несколько кубов, и эти кубы известны как гиперкубы.

6.3. Операции OLAP:

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

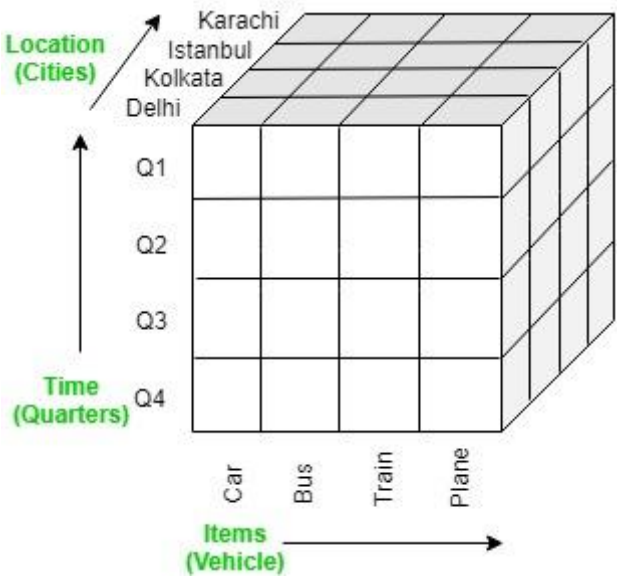

Рисунок 26. Olap куб

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

- Спуск по иерархии концепций
- Добавление нового измерения

В кубе, приведенном в разделе обзора, операция детализации выполняется путем перемещения вниз в иерархии понятий измерения «Время» (Квартал -> Месяц).

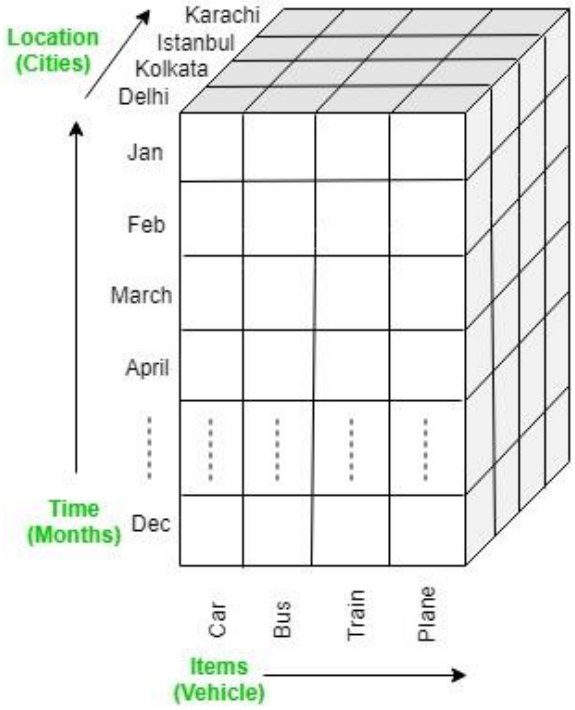

Рисунок 27. Пример OLAP операции Детализация (Drill down)

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

- Восхождение в иерархии концепций
- Уменьшение размерности

В кубе, приведенном в разделе обзора, операция сворачивания выполняется путем подъема вверх по иерархии понятий измерения «Местоположение» (Город -> Страна).

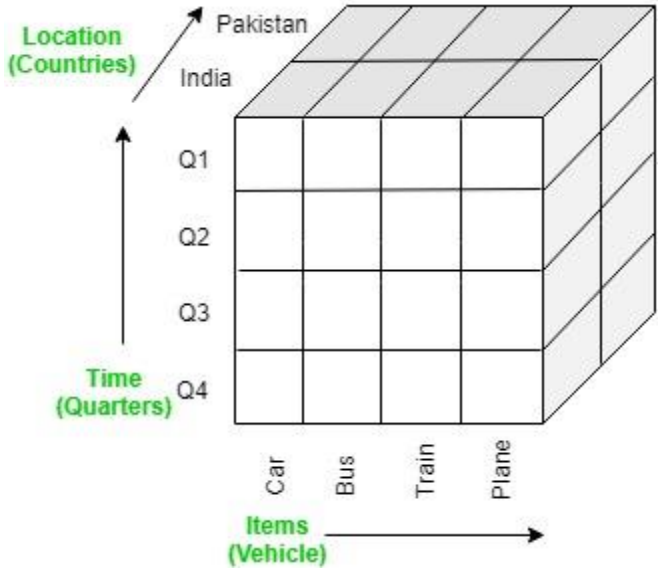

Рисунок 28. Пример OLAP операции Сворачивание (Roll up)

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»

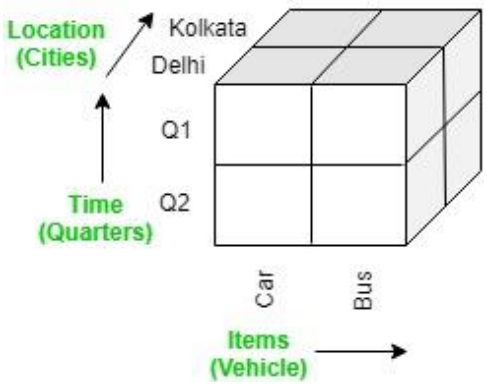

Рисунок 29. Пример OLAP операции Вырезы (Dice)

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».

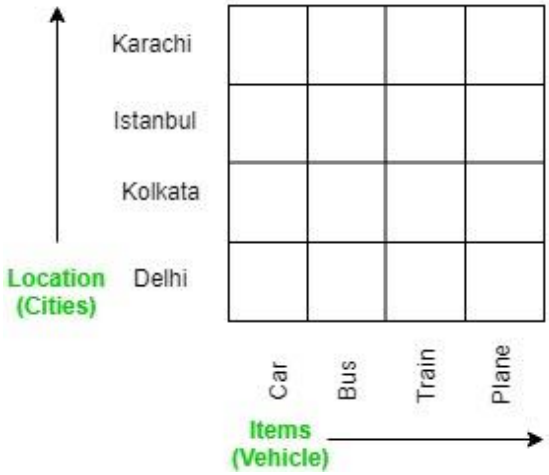

Рисунок 30. Пример OLAP операции Срез(slice).

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

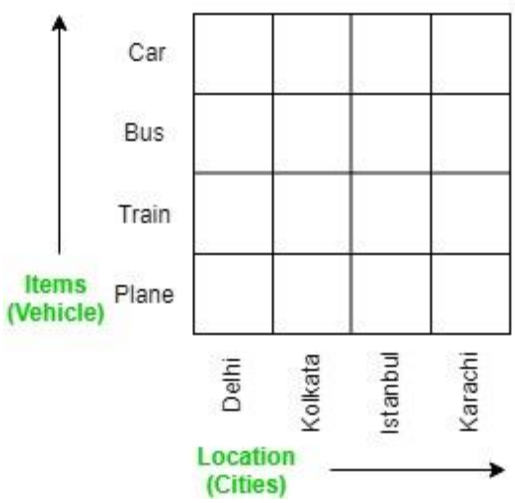

Рисунок 31. Пример OLAP Операции поворота.

При работе с OLAP-системой, всегда можно найти ответы, на возникающие вопросы, увидеть картину в целом, проводить постоянный мониторинг состояния бизнеса. При этом можно быть уверенными, что используется только актуальная информация.

OLAP-кубы содержат бизнес-показатели, используемые для анализа и принятия управленческих решений, например: прибыль, рентабельность продукции, совокупные средства (активы), собственные средства, заемные средства и т.д.

Благодаря детальному структурированию информации OLAP-кубы позволяют оперативно осуществлять анализ данных и формировать отчёты в различных разрезах и с произвольной глубиной детализации. Отчёты могут создаваться аналитиками, менеджерами, финансистами, руководителями подразделений в интерактивном режиме для того, чтобы быстро получить ответы, на возникающие ежедневно вопросы, и принять правильное решение. При этом сотрудникам, для создания отчетов не нужно прибегать к услугам программистов, на что обычно уходит немало времени.

	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 требованиям.	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

Таблица 6. Сравнение функциональных и эксплуатационных характеристик OLAP и OLTP концепций

7. Концепция корпоративных хранилищ данных (КХД). (Озера данных, витрины данных, Minimum Viable Platform)

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

7.1 Хранилище данных (Datawarehouse, DWH)

Хранилище данных (Datawarehouse, DWH) — это центральный репозиторий информации, которую можно анализировать для принятия более обоснованных решений. Данные поступают в хранилище из транзакционных систем, реляционных баз данных и других источников — как правило, с определенной периодичностью. Бизнес-аналитики, специалисты по работе с данными и лица, ответственные за принятие решений, получают доступ к данным с помощью инструментов бизнес-аналитики, SQL-клиентов и других приложений для аналитики.

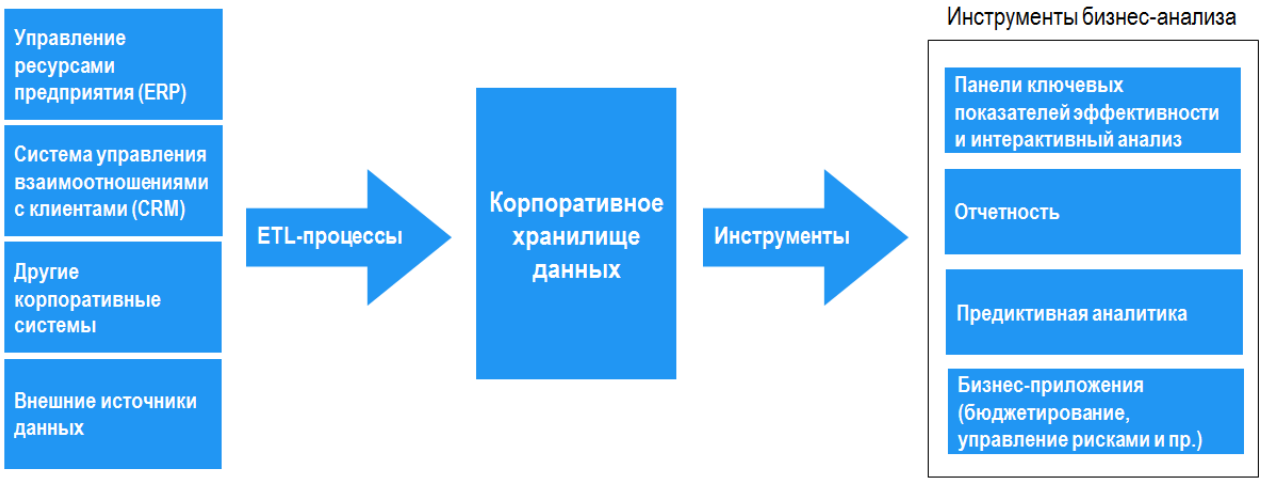

Рисунок 32. Функциональная схема Корпоративного Хранилища Данных.

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:

поддержка 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.

Удовлетворять всем перечисленным требованиям в рамках одного и того же продукта зачастую не удается. Поэтому для реализации хранилищ данных обычно используется несколько продуктов, одни их, которых представляют собой собственно средства хранения данных, другие — средства их извлечения и просмотра, третьи — средства их пополнения и т.д.

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных. Как правило, эта структура денормализована (это позволяет повысить скорость выполнения запросов), поэтому может допускать избыточность данных.

Во-первых, обычные базы данных предназначены для того, чтобы помочь пользователям выполнять повседневную работу, тогда как хранилища данных предназначены для принятия решений. Например, продажа товара и выписка счета производятся с использованием базы данных, предназначенной для обработки транзакций, а анализ динамики продаж за несколько лет, позволяющий спланировать работу с поставщиками, — с помощью хранилища данных.

Во-вторых, обычные базы данных подвержены постоянным изменениям в процессе работы пользователей, а хранилище данных относительно стабильно: данные в нем обычно обновляются согласно расписанию (например, еженедельно, ежедневно или ежечасно — в зависимости от потребностей). В идеале процесс пополнения представляет собой просто добавление новых данных за определенный период времени без изменения прежней информации, уже находящейся в хранилище.

И в-третьих, обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.

Как мы уже знаем, конечной целью использования OLAP является анализ данных и представление результатов этого анализа в виде, удобном для восприятия и принятия решений. Основная идея OLAP заключается в построении многомерных кубов, которые будут доступны для пользовательских запросов. Однако исходные данные для построения OLAP-кубов обычно хранятся в реляционных базах данных. Нередко это специализированные реляционные базы данных, называемые также хранилищами данных (Data Warehouse). В отличие от так называемых оперативных баз данных, с которыми работают приложения, модифицирующие данные, хранилища данных предназначены исключительно для обработки и анализа информации, поэтому проектируются они таким образом, чтобы время выполнения запросов к ним было минимальным. Обычно данные копируются в хранилище из оперативных баз данных согласно определенному расписанию.

7.2 Озера Данных

Озеро данных, или (Data Lake) — коллекция инстансов для хранения различных типов данных в дополнение непосредственно к источникам данных

Представляет собой хранилище 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)

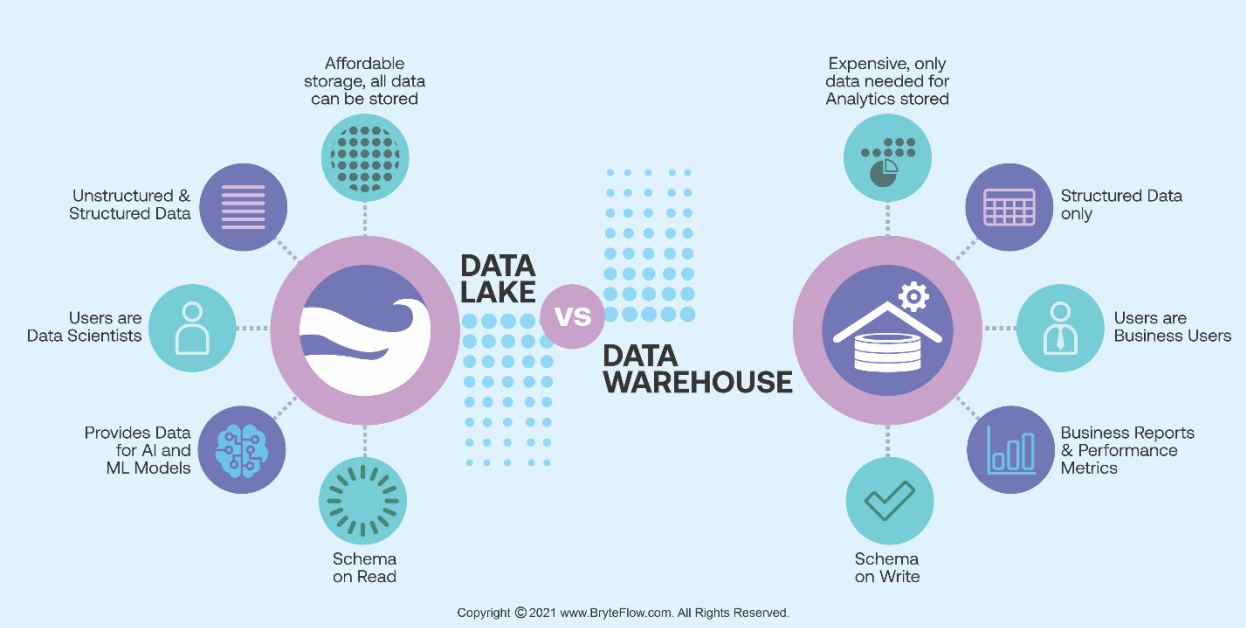

Рисунок 33. Сравнение функциональных характеристик Озера данных и КХД

- Загружать данные в максимальном удобном виде для аналитических приложений;
- В процессе загрузки данных обогащать их дополнительной информацией;
- Фиксировать и документировать lineage (происхождение) данных.

7.3 Схема на чтение против схемы на запись

Шаблон на запись (Schema-on-Write). Эта конструкция тесно связана с управлением 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 по мере загрузки данных.

Сегодня данные и инструменты аналитики незаменимы для компаний, которые стремятся сохранять преимущества перед конкурентами. Чтобы превращать данные в полезную аналитическую информацию, следить за эффективностью ведения бизнеса и принимать обоснованные решения, компании используют отчеты, панели управления и различные аналитические инструменты. За этими отчетами, панелями управления и аналитическими инструментами стоят хранилища данных, которые эффективно хранят данные, минимизируя количество операций чтения и записи и быстро возвращая результаты запросов сотням и тысячам пользователей одновременно.

7.4. Концепции единая версия правды SSOT и Единая версия истины SVOT

При моделировании DWH обычно подразумеваются концепции single version of truth единая версия правды) и historical truth (историческая правда)

Единый источник истины (SSOT) по сути является синонимом SSOD. Однако, когда упоминается термин «истина», обычно речь идет о хранилище данных, потому что, хотя истина предполагается, когда элемент данных может быть найден только в одном месте, это предположение не существует, когда элемент

данных может можно найти в нескольких местах. Различие между SSOT и SVOT заключается в том, что первый из них может включать ссылки на элементы данных по ссылке в других регионах. Когда такие элементы данных обновляются, распространение инициируется по предприятию, поэтому повторяющиеся элементы данных всегда обновляются, что приводит к одной и той же «истине» независимо от того, какой элемент впоследствии считывается процессами.

Теперь, когда мы все дальше отходим от представленной исходной концепции, SSOD, появляется больше вариаций методов достижения истины, но концептуально SSOT похож на третью диаграмму ниже. Как и в случае с SVOT, дополнительная логика располагается поверх данных для устранения различий, но в исходных данных не существует нескольких версий истины. В этом случае, когда истина для определенного элемента данных обновляется, инициируется логика для распространения этих данных на другие, локали, где этот элемент данных также может храниться, так что, хотя есть единственный источник истины, эта истина может существовать в несколько мест, где к нему может получить доступ произвольный бизнес-процесс.

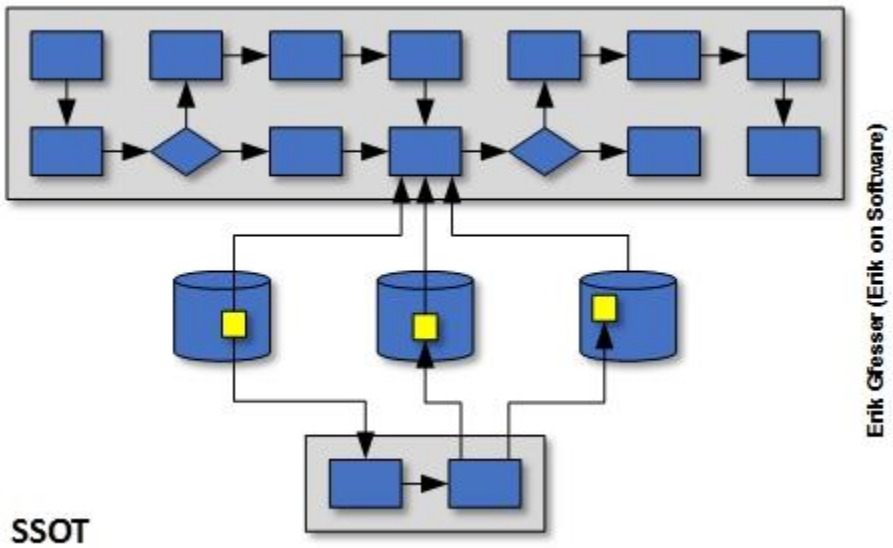

Рисунок 34. Схема движения потоков данных SSOT

SVOT

Единая версия истины (SVOT) обычно указывает на то, что в организации может существовать несколько версий истины. Одна из целей инициативы по созданию хранилищ данных - раскрыть эти версии правды и предоставить организации «официальную» версию правды. Одна из распространенных причин, по которой может существовать несколько версий истины, заключается в том, что такие данные могут храниться в организационных разрозненных хранилищах. Билл

Инмон, «отец хранилищ данных», красноречиво описал эту концепцию в материалах, которые он писал на протяжении многих лет. Концептуально это выглядит примерно, как вторая диаграмма ниже. Вы заметите, что истина, к которой должен получить доступ произвольный процесс, может быть найдена только в одном месте, как и в случае с SSOD, но с SVOT существует дополнительная логика между этой истиной и исходными базами данных. Эта логика разрешает любые споры, которые эти отдельные транзакционные базы данных могут иметь друг с другом, и решение сохраняется в хранилище данных для использования бизнес-процессом.

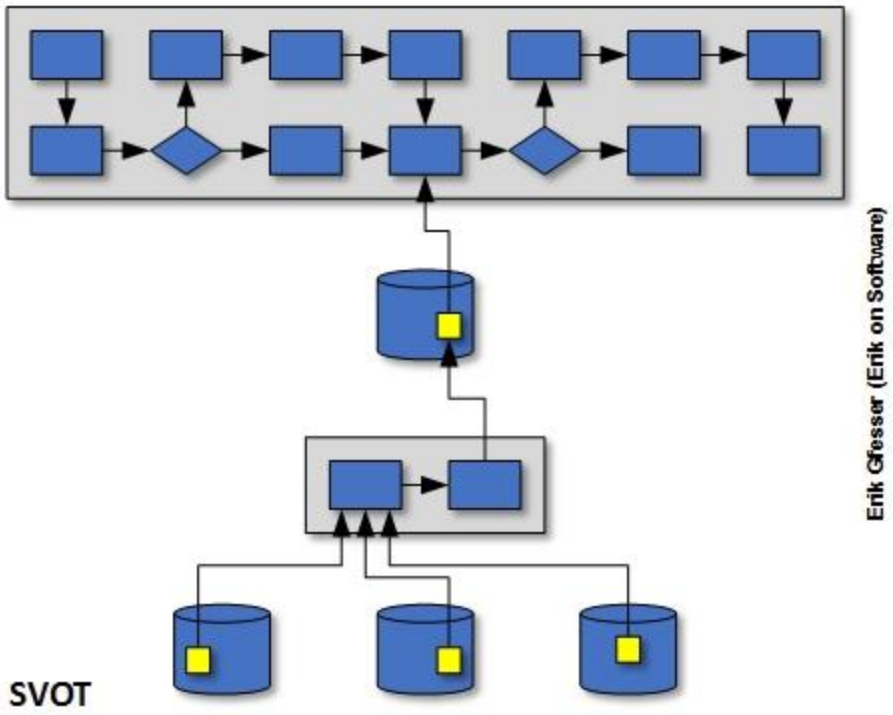

Рисунок. Схема движения потоков данных SVOT

Небольшие группы обычно начинают с ручного управления задачами, такими как очистка данных, обучение моделей машинного обучения, отслеживание результатов и развертывание моделей на производственном сервере. По мере роста размера команды и решения растет и количество повторяющихся шагов. Также становится более важным, чтобы эти задачи выполнялись надежно. Сложность зависимости этих задач друг от друга также увеличивается. Когда вы начинаете, у вас может быть конвейер задач, которые нужно выполнять раз в неделю или раз в месяц. Эти задачи необходимо запускать в определенном порядке. По мере вашего роста этот конвейер становится ** сетью ** с динамическими ветвями. В некоторых случаях некоторые задачи запускают другие задачи, и это может зависеть от нескольких других задач, запущенных в первую очередь.

8. Конвейеры ETL и ELT

Обслуживание данных включает в себя такие действия с данными, как перемещение, интеграция, очистка, обогащение и etl-процессы (extract, transform, load).

8.1 Процессы ETL (Extract, Transform, Load)

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 включает следующие этапы:

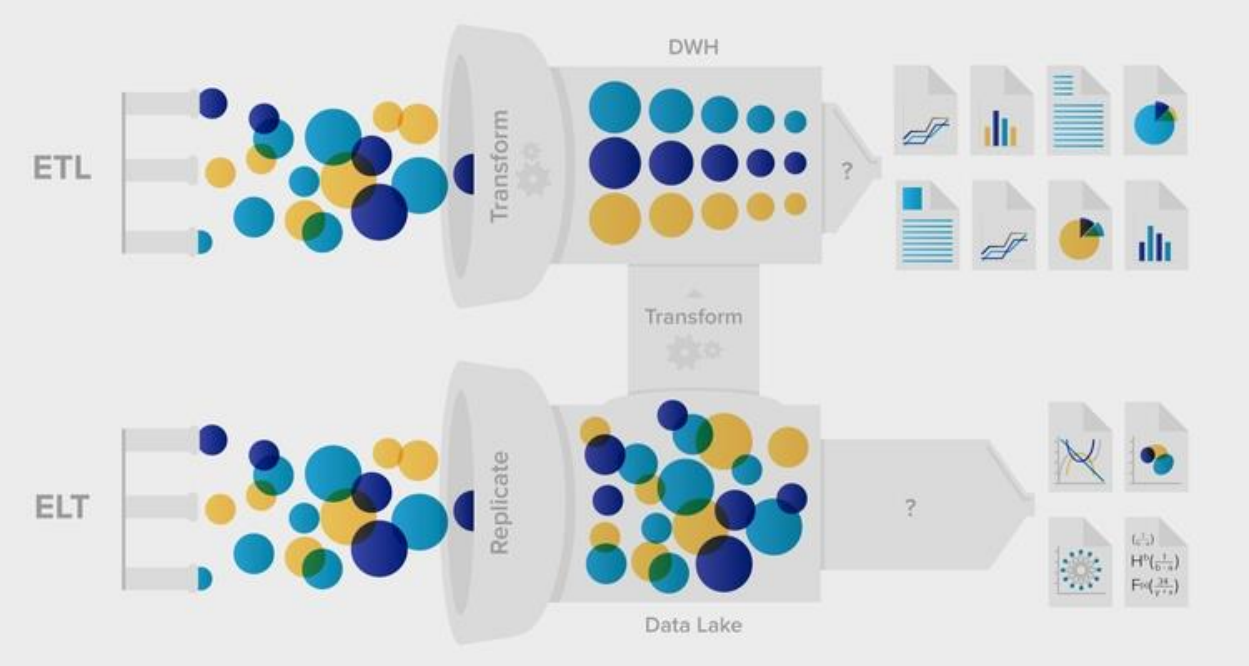

Рисунок 35. Метамодель ETL и ELT в прикладных задачах

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.

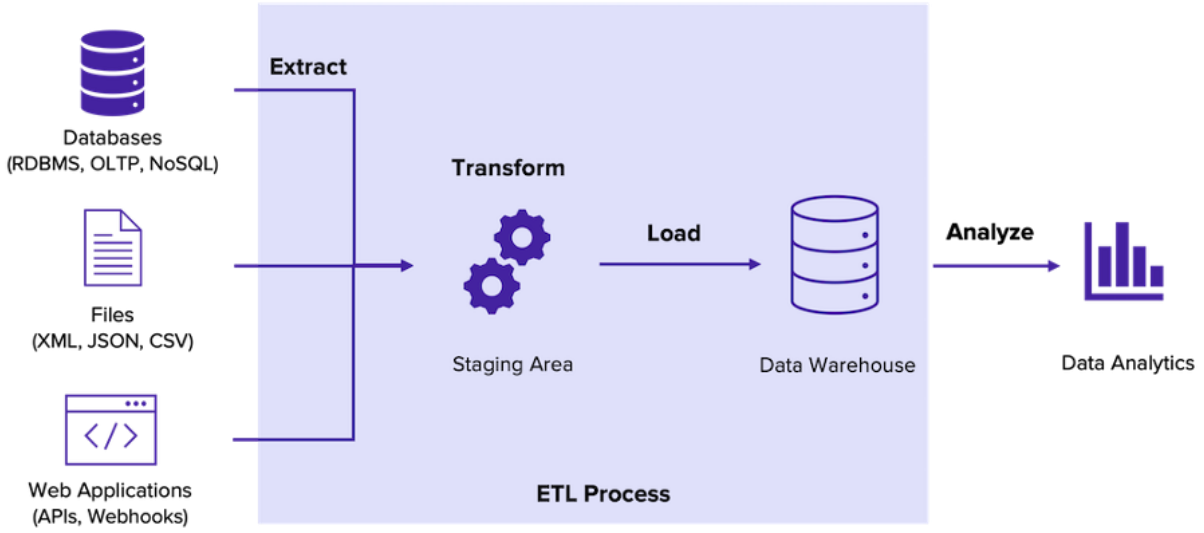

Рисунок 36. Структурная схема конвейера ETL

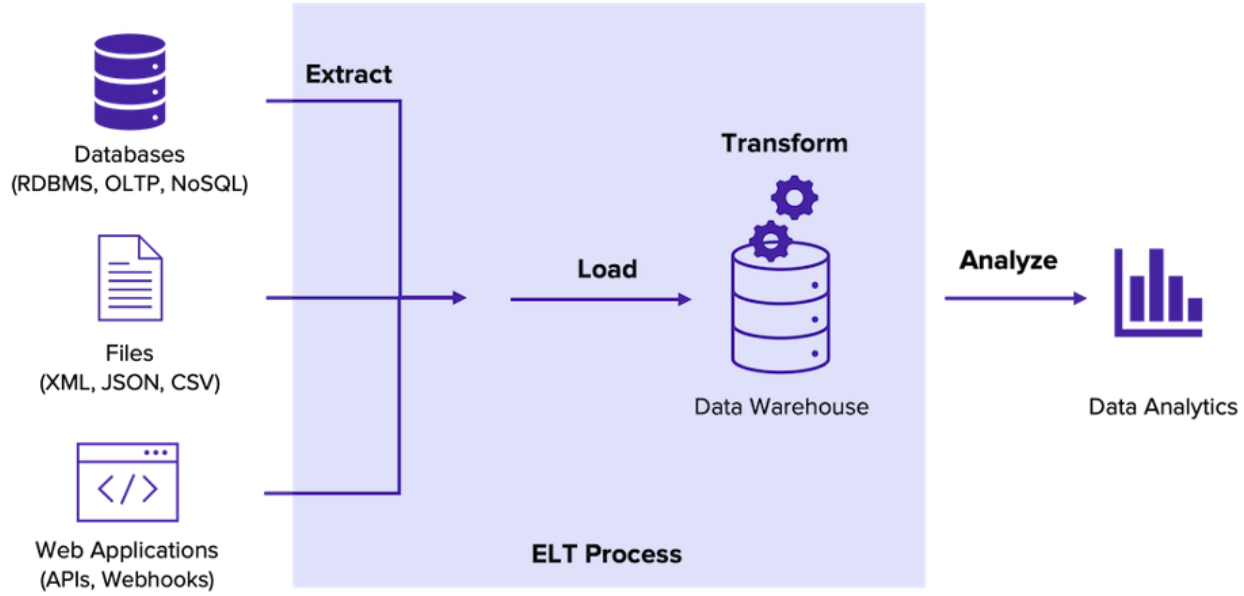

Рисунок 37. Структурная схема конвейера ELT

	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	В процессе ELT преобразование данных выполняется по мере необходимости в самой целевой системе. В результате этап преобразования занимает мало времени, но может

	преобразование больших наборов данных может занять много времени, но анализ может выполняться сразу после завершения процесса ETL.	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	Учитывая, что процесс ELT изначально основан на облаке, дополнительное оборудование не требуется.

	облачные решения ETL не требуют оборудования.	
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.

Таблица 7. ETL и ELT - 10 ключевых отличий:

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	Синхронизация данных между различными процессами, системами и структурами	Управляемое озеро данных

количественных данных	предприятия автоматизирована не полностью, часть процедур запускается по требованию или вручную	
5. Оптимизируемый	Процедуры автоматизированного появления, обновления, обмена и синхронизации данных между различными процессами, системами и структурами предприятия отлажены и успешно работают	Самоорганизующееся озеро данных

Таблица 8. Стадии становления Data Lake

Примеры использования озера данных

Здравоохранение. Хранилища данных уже много лет используются в сфере здравоохранения. 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.

8.2 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. Это 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 продвижения и рекламных кампаний, на основе их отзывов и моделей поездок. Они также используют DW для выполнения повседневных операций.

9. ФОРМАТЫ ФАЙЛОВ ХРАНЕНИЯ BIG DATA

9.1 Базовые схемы хранения данных в парадигме NoSQL и их основные характеристики.

В простейшем случае это так называемый открытый текст: CSV, XML, JSON, JSONB, YAM, BLOB. Данные, хранящиеся в этом формате, в основном форматируются таким образом, что поля имеют фиксированную ширину или разделитель. Отдельные записи в формате CSV разделяются запятыми. XML. Можно использовать определения внешних схем в XML. Однако производительность сериализации и десериализация, как правило, является плохой. JSON. Нотация объектов JavaScript (JSON) эффективнее XML, но существует проблема с производительностью сериализации и десериализации.

Использование соответствующего формата файла должно привнести следующие преимущества для системы:

1. Время чтения уменьшается.
2. Время записи сокращается.
3. Файлы могут быть разделены, что, другими словами, означает, что больше нет необходимости читать весь файл для получения меньшего его подраздела.
4. Существует поддержка эволюции редактирования схемы, и схема может быть изменена по запросу в зависимости от изменения потребностей системы.
5. Имеются усовершенствованные кодеки сжатия для обеспечения возможности сжатия файлов без потери преимущества базового формата.

9.2 Специализированные форматы линейные (строковые) и колоночные (столбцовые).

LOGICAL TABLE STRUCTURE

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

Рисунок 38. Пример типовой таблицы

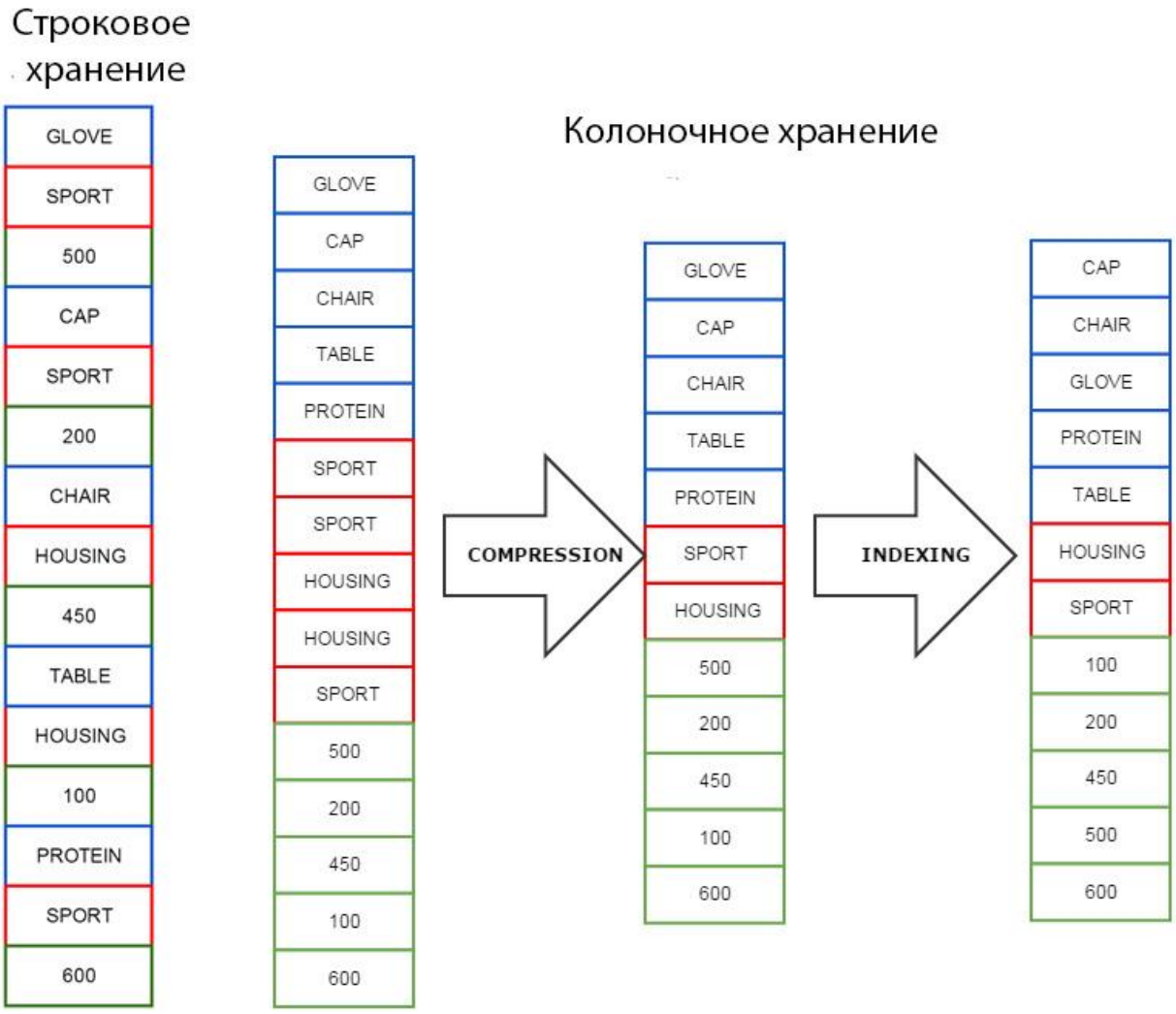

Рисунок 39. Физическая структура размещения блоков данных для строковых и колоночных данных.

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

Таблица 8. Функциональные различия между хранилищем строк и хранилищем столбцов

9.3. Конвейеры распараллеливания передачи блоков данных. Сериализация.

Сериализация — это процесс преобразования данных из текстового формата в двоичный, необходимый для передачи данных по сети и сохранения информации в виде файла на диске, в памяти или базе данных.

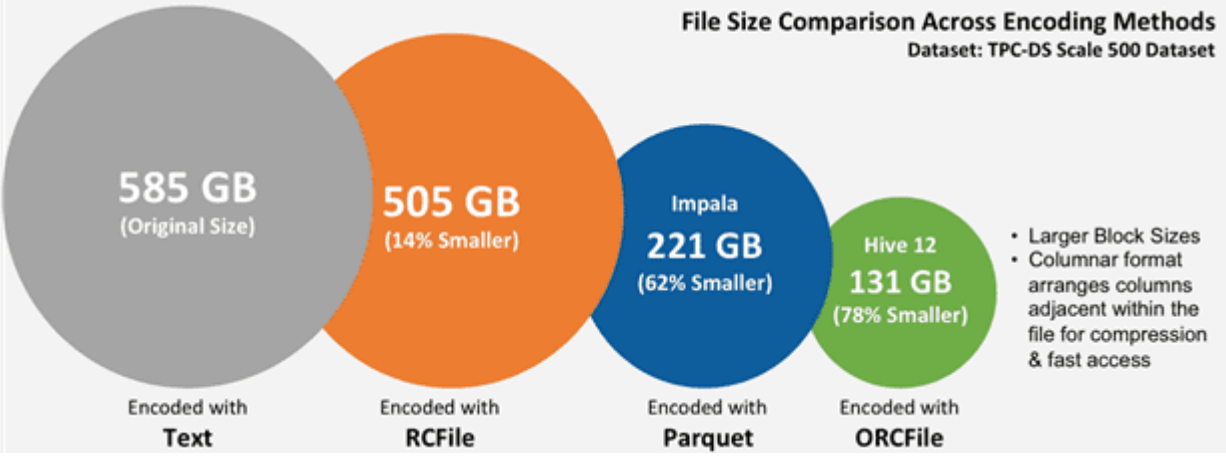

Рисунок. 40

Avro — обеспечивает высокую скорость записи информации, и потому отлично подходит для обработки потоков Big Data в Apache Kafka, Flume и корпоративных озерах данных (Data Lake), а также хорошо решает задачи полного

чтения всех полей записи, что требуется в ETL-хранилищах и витрин данных. в основном позиционируется как замена Thrift: он не требует генерации кода, может передавать схему вместе с данными или вообще работать с динамически типизированными объектами. Способен абстрагироваться от изменений схемы с помощью управления версиями (механизм форматирования, а не VCS) или даже с помощью реестра схемы.

Parquet — колончатый формат, оптимизированный для хранения сложных структур и эффективного сжатия. Изначально был разработан в Twitter, а сейчас является одним из основных форматов в инфраструктуре Hadoop (в частности, его активно поддерживают Spark и Impala). Apache Impala, Drill и Big Data платформе MAPR.

ORC — новый оптимизированный формат хранения данных для Hive.

- индексация блоков каждого столбца, что ускоряет операции ввода-вывода;
- считывание метаданных на уровне столбца позволяет оптимизировать SQL-запросы;
- поддержка ACID-требований к транзакциям (Atomicity — Атомарность, Consistency — Согласованность, Isolation — Изолированность, Durability — Долговечность);
- более высокая степень сжатия файлов экономит место на жестком диске

CarbonData - Этот формат данных был разработан компанией Huawei для устранения существующих недостатков в уже имеющихся форматах. CarbonData – это относительно новый формат файла данных, который позволяет разработчикам использовать преимущества формата, ориентированного на столбцы, но при этом иметь возможность обработки запросов произвольного доступа. Данные сгруппированы в блокноты, т.е. хранятся наряду с другой информацией о данных, такой как схема, индексы и смещения [3]. Метаданные хранятся в верхних и нижних колонтитулах, что обеспечивает значительную оптимизацию производительности во время сканирования и обработки

Thrift — эффективный, но не очень удобный бинарный формат передачи данных. Работа с этим форматом предполагает определение схемы данных и генерацию соответствующего кода клиента на нужном языке, что не всегда возможно. В последнее время от него стали отказываться, но многие сервисы всё ещё используют его.

Feather (import pyarrow.feather as feather), msgpack, sequence

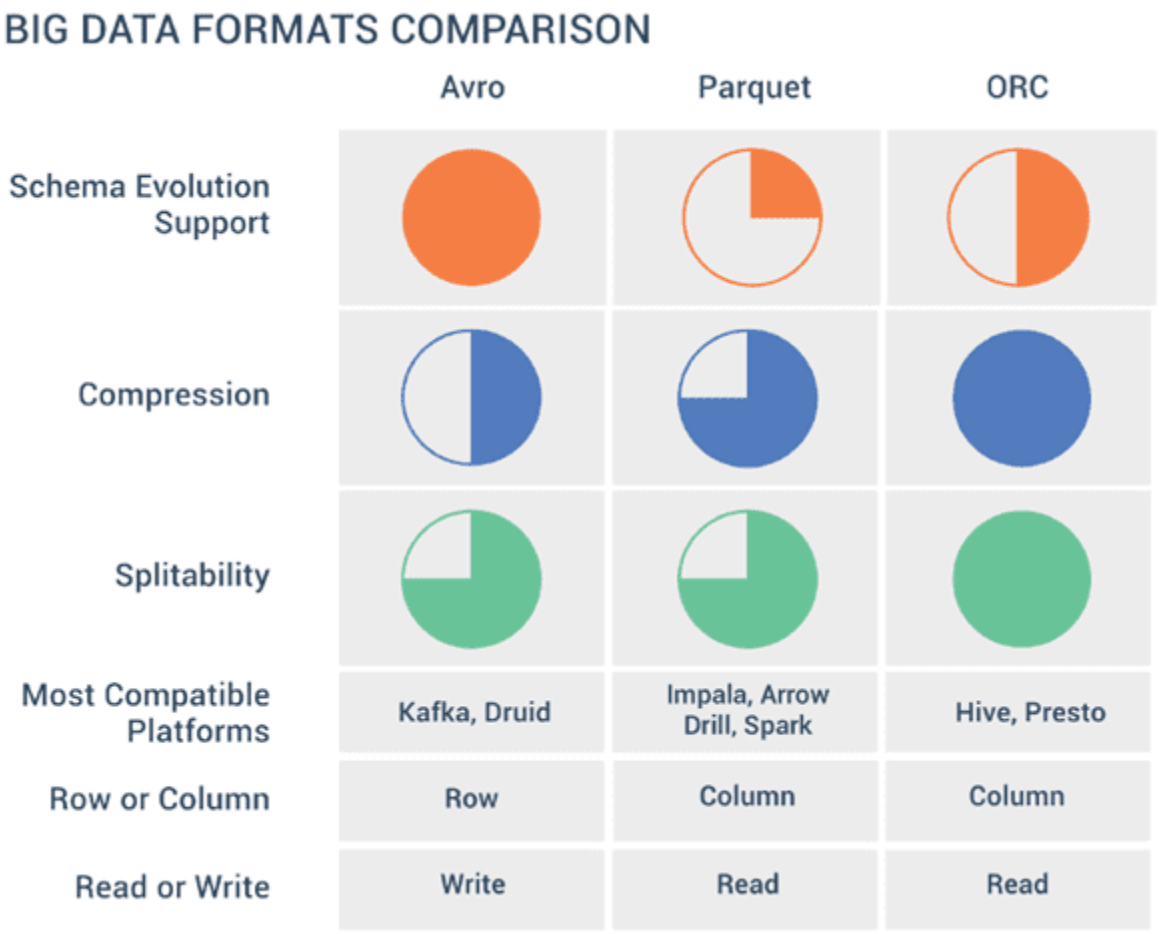

Рисунок 41. Форматы AVRO, Parquet, ORC

Критерий	**Текстовый**	**Sequence**	**Avro**	**Parquet**	**ORC**
Тип	Построчный	построчный	построчный	колоночный	Комбинированный
Человекочитаемость	да	нет	нет	нет	нет
Самоописательный	нет	нет	да	да	да
Сложные типы	нет	да	да	да (ограничено)	да
Разделяемость	Нет (иногда)	да	да	да	да
Сжатие	нет	да	да	да	да

Эффективность запросов	низкая	низкая	средняя	высокая	высокая
Индекс и статистика и т. д.	нет	нет	нет	нет	ACID, индексы, статистика
Сериализация	нет	нет	да	нет	нет
Схема	нет	нет	да, хранится в самом файле	да	да
Развитие схем	нет	нет	да	да	Да (в новых версиях)
Эффективность записи	высокая	высокая	Средне-высокая	низкая	низкая
Эффективность чтения	низкая	средняя	Средне-низкая	высокая	высокая

Таблица 9. Сравнительные характеристики основных форматов хранения Больших Данных