

Data Warehouse Analyst



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Аналитические движки (СУБД) для работы с данными



Алексей Железной

Senior Data Engineer в Wildberries
Магистратура - ФКН ВШЭ

Руководитель курсов **DWH Analyst, ClickHouse** для инженеров и архитекторов
БД в OTUS

Преподаватель курсов **Data Engineer, DWH Analyst, PostgreSQL** и пр. в OTUS

[LinkedIn](#)

Правила вебинара



Активно
участвуем



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Знакомство в чате группы

1. Где работаете? Чем занимаетесь?
2. Какие из тем знакомы? Уровень владения? На чем хотелось бы сделать акцент?
3. Ожидания от курса в целом?

Маршрут вебинара



Аналитические СУБД как класс

Ключевые принципы устройства аналитических СУБД

Лучшие практики использования аналитических движков

Демо и рассмотрение конкретных примеров

Рефлексия

Цели вебинара

1. Провести знакомство с курсом, платформой и планом работы
 2. Разобрать примеры аналитических движков и их использование на практике
-

Аналитические СУБД как класс

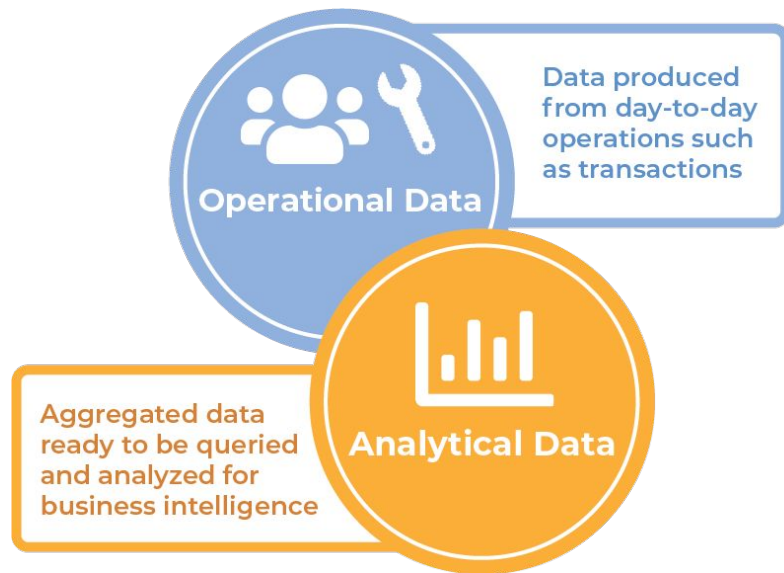
Инфраструктура данных включает

Operational Data Systems

- Операционные данные — это данные, которые производятся в ходе повседневной деятельности вашей организации.
- Поддерживают доступ к OLTP.

Analytical Data Systems

- Аналитические данные используются для принятия бизнес-решений
- Аналитические данные лучше всего хранить в системе данных OLAP

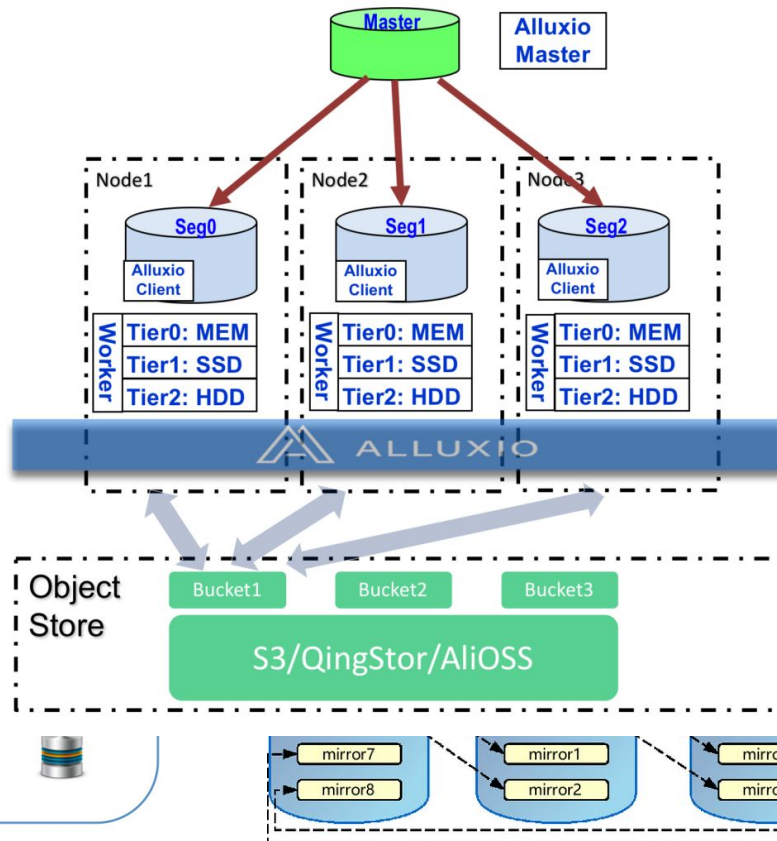
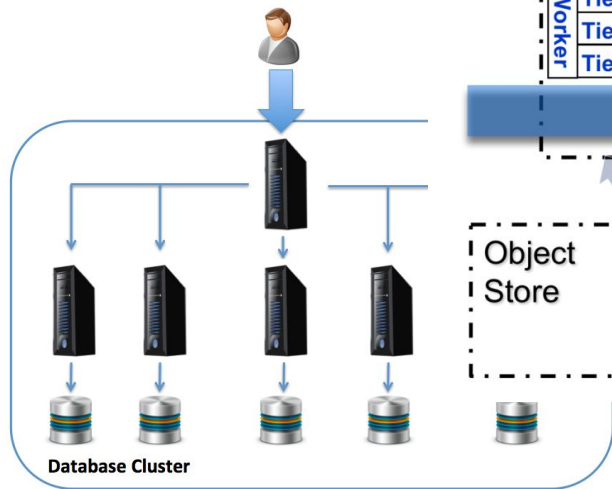


Теория аналитических СУБД

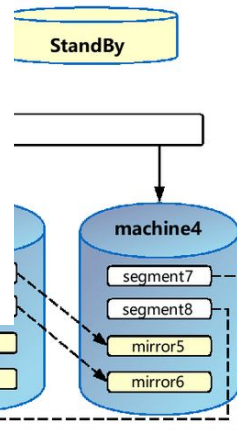
Что такое MPP, и как это работает?

- Massive Parallel
- Выделение неограниченного объема памяти

MPP System Architect

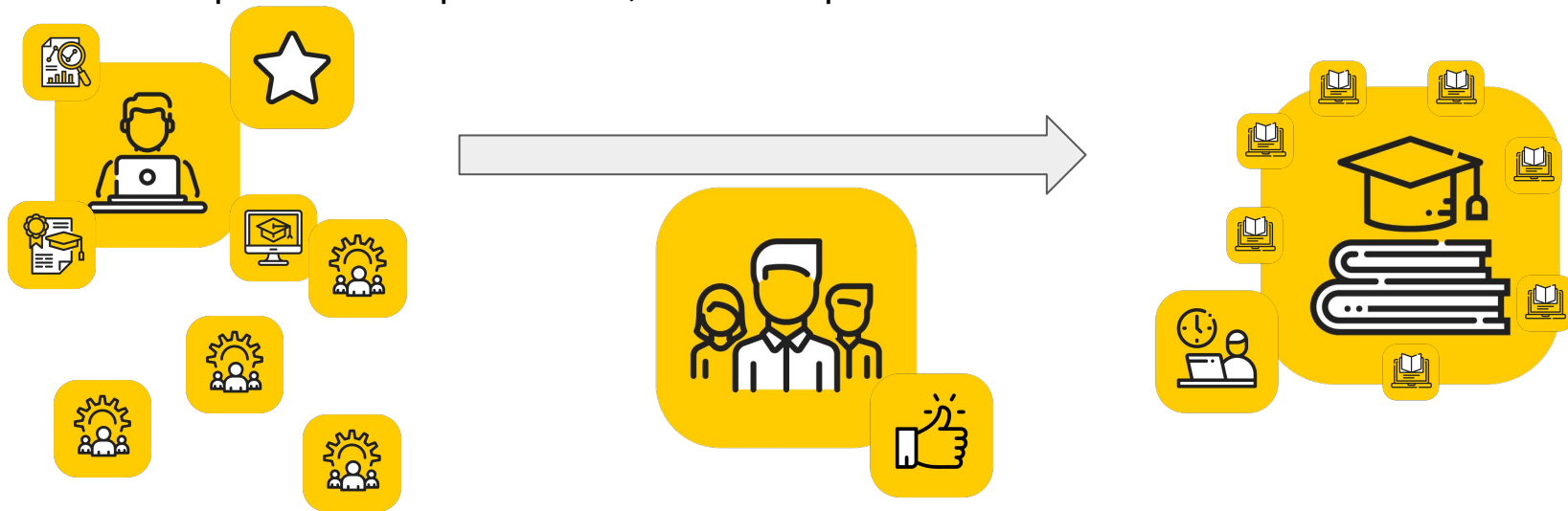


отка данных)
ольким различным



Анализ больших данных: человеческий пример

Масштабировать по горизонтали, а не по вертикали

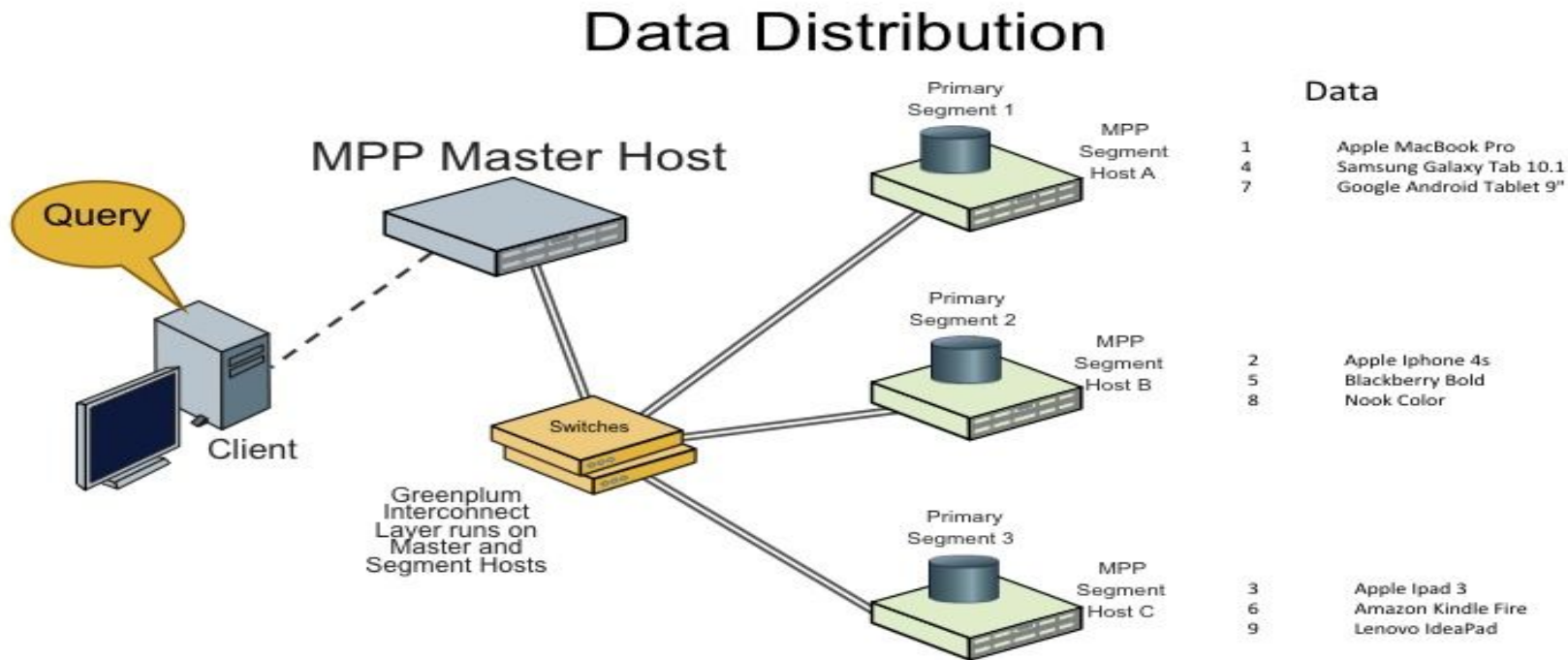


Это массивно-параллельная обработка в действии, только с людьми, а не с компьютерами. Разделение простых, но больших задач на несколько сегментов и одновременная обработка этих сегментов будет намного быстрее, чем один человек, работающий в одиночку, независимо от того, насколько он квалифицирован.

Что такое MPP, и как это работает?

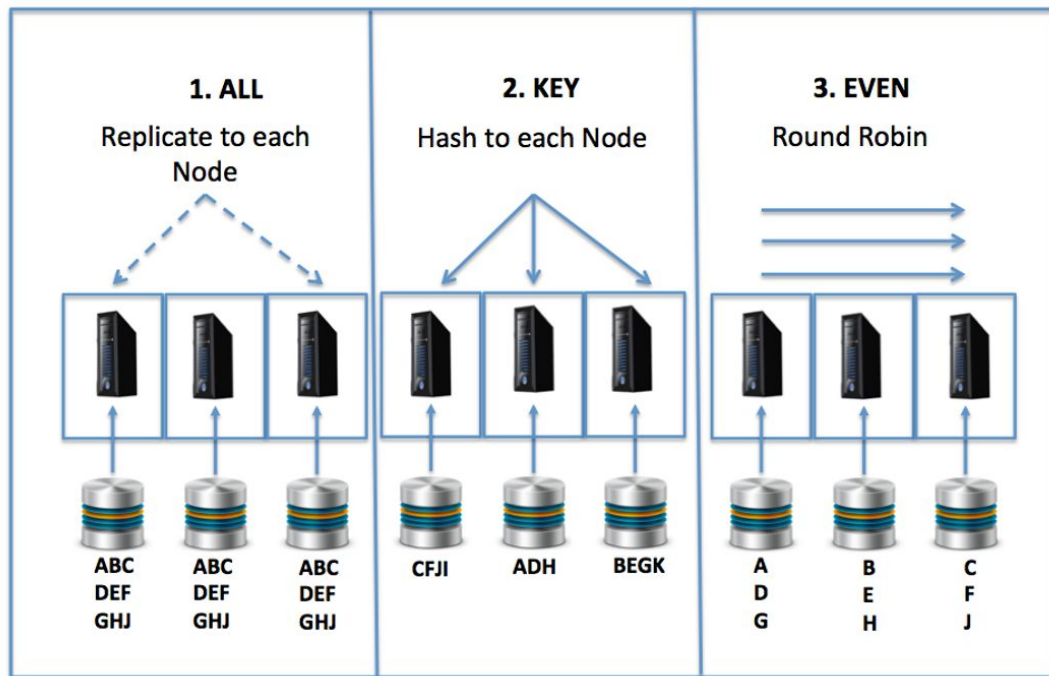
- **База данных MPP** — это тип базы данных или хранилища данных, в котором данные и вычислительная мощность распределяются между несколькими различными узлами (серверами) с одним ведущим узлом и одним или несколькими вычислительными узлами
- В MPP лидера (вас) называют ведущим узлом, работников библиотеки - вычислительными узлами
- Базы данных MPP можно масштабировать горизонтально, добавляя дополнительные вычислительные ресурсы (узлы)
- **Плюсы** — относительная быстрота обработки больших объемов данных (*Shared Nothing*), простота горизонтального масштабирования до сотен узлов, отказоустойчивость за счет зеркалирования и резервирования.
- **Минусы** — высокие требования к ресурсам, низкая производительность при большом объеме простых запросов, неоптимальное распределение сегментов

Что такое MPP, и как это работает?



MPP Data Distribution

Three MPP Data Distribution Styles



Представители MPP СУБД



Google BigQuery



Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

OLAP - просто о сложном


OLAP

OLAP = OnLine Analytical Processing = аналитическая обработка данных в реальном времени. *Многомерные БД*

O L A P

Processing = Обработываются некие исходные данные...

Analytical = ... чтобы получить какие-то аналитические отчеты или новые знания...



OnLine = ... в реальном времени, практически без задержек на обработку.

Бизнес-смысл OLAP

Залог успешного бизнеса с точки зрения Big Data:

- +** Много данных (как фактовых, так и исторических)
 - +** Проработанные механизмы сбора и обработки данных
 - +** Крутые, мощные системы для их хранения, анализа
 - +** Наглядные BI-системы для отображения информации
- ☐ Принятие подготовленных, “правильных” бизнес-решений

Сферы применения - в анализе тенденций, финансовой отчетности, прогнозировании продаж, бюджетировании и других целях планирования

Тех-смысл OLAP

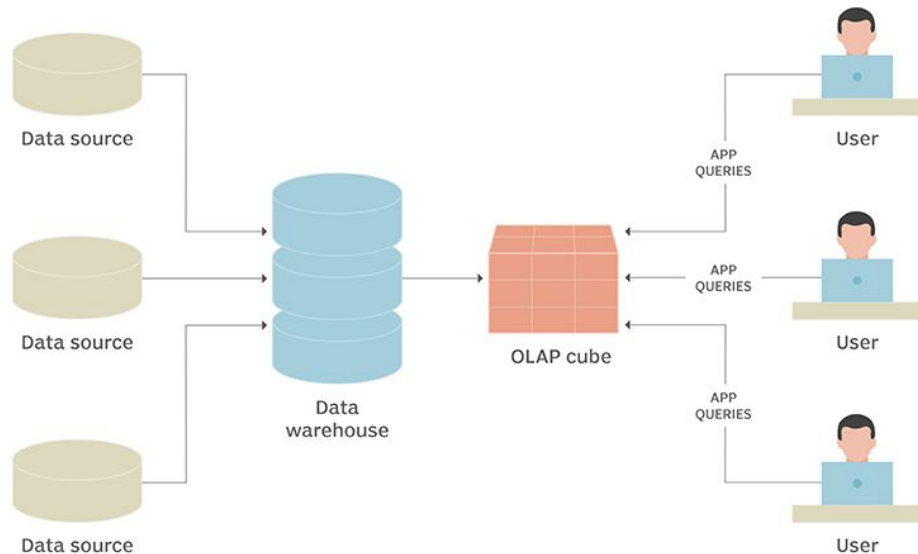
- используется для построения отчетов на основе больших объемов накопленных исторических данных за огромные промежутки времени, но эти отчеты обновляются не слишком часто (*)
- чаще всего это столбцовые СУБД (или поддерживающие column-orientation)
- выбирает данные быстро
- в центре находится таблица фактов, в которой находятся все показатели (сумма, кол-во) и ссылки на справочники (*)
- чем больше столбцов, тем ниже скорость выполнения операций над строками (таких как добавление или изменение данных)
- больше про денормализацию

Как работают системы OLAP

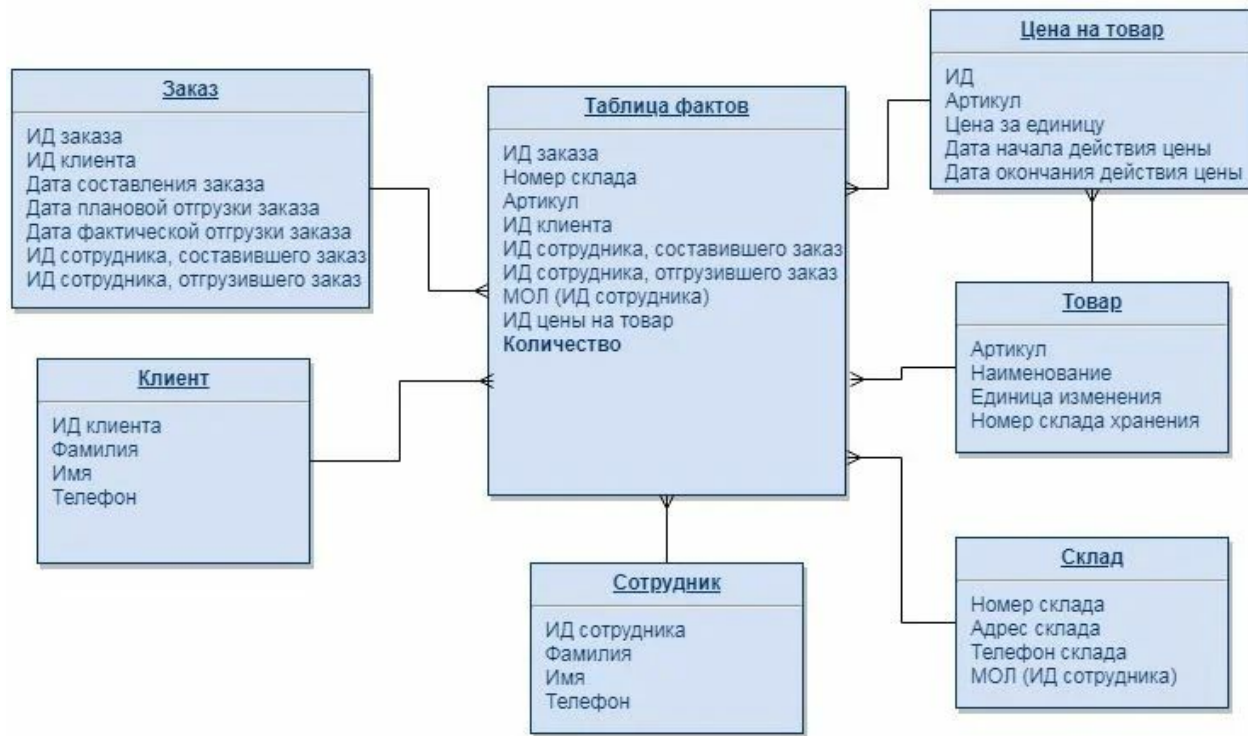
- **Roll-up** - обобщение данных по измерениям
- **Drill-down** - углубление во временном периоде
- **Slice** - выбор определенного уровня отображения
- **Dice** - выбор данных из нескольких измерений
- **Pivot** - поворот осей данных куба

The OLAP process

How data is prepared for online analytical processing (OLAP)



Примеры OLAP



	Март		
	Февраль		
	Январь		
	США	Канада	Мексика
Напитки	10 000	2000	1 000
Продукты питания	5000	500	250
Прочие товары	5000	500	250

Виды OLAP-систем

- **ROLAP = Relational OLAP (построенный на отношениях таблиц и баз данных между собой)**
 - система, которая напрямую ничего не хранит, но умеет быстро вынимать все
 - данные разложены по однотипным СУБД (PostgreSQL)
 - достаточно редко встречается (либо часто, но в маленьких компаниях)
- **MOLAP = Multidimensional OLAP ()**
 - данные лежат не только в однотипных корпоративных базах данных (надо все собирать и складывать вместе)
 - предварительная подготовка данных
 - позволяет структурировать данные под разные запросы пользователей
- **Гибрид первых двух систем**

Многомерные БД для OLAP



Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет



OLTP - несложно о простом

OLTP

OLTP = OnLine Transactional Processing - обработка транзакций в реальном времени.

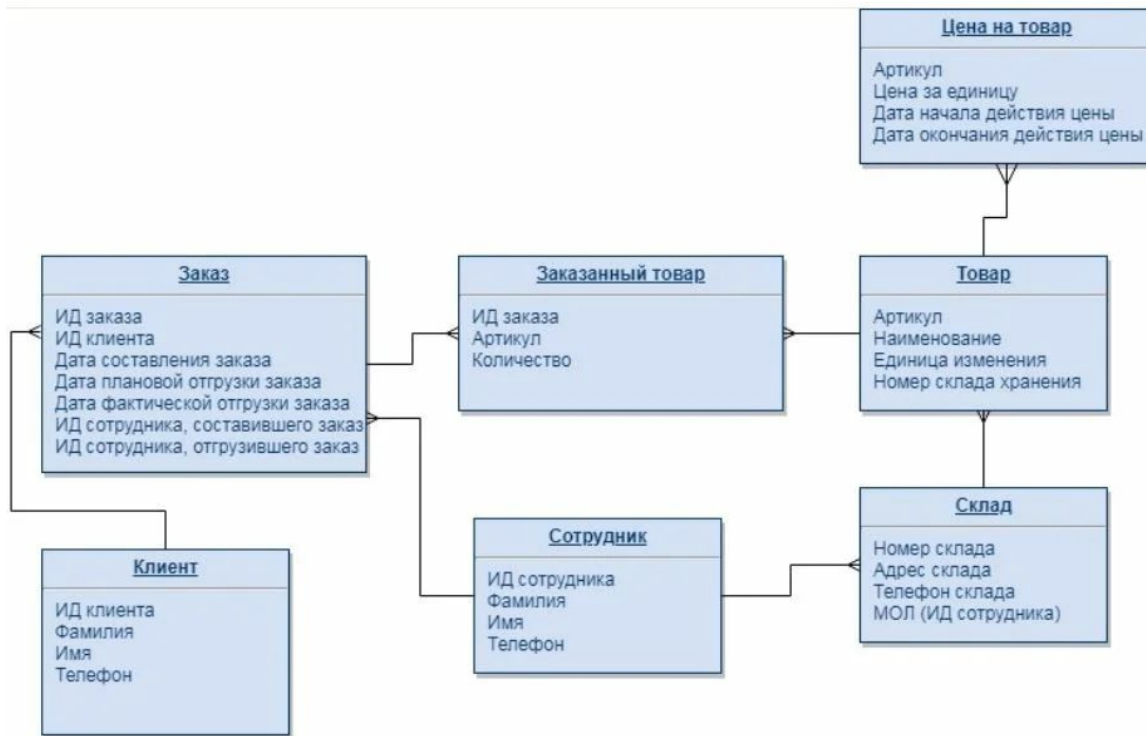
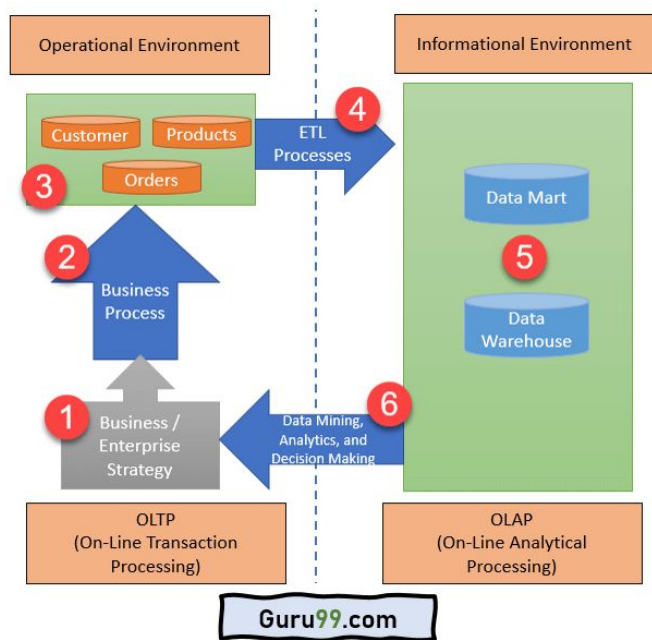
Реляционные БД

- Одновременное выполнение нескольких транзакций (экономических, финансовых, цифрового взаимодействия), таких как интернет-банкинг, покупки, ввод заказов или отправка текстовых сообщений
- Задача - ввод, редактирование, удаление данных в режиме онлайн и их хранение
- Больше про нормализацию

Особенности OLTP

- Нормализованные данные;
- Высокая интенсивность добавления и изменения данных;
- Большое количество одновременно активных пользователей (*)
- Внесение данных и расчеты осуществляют пользователи системы;
- Содержат актуальные данные (*)
- OLTP использует транзакции, которые включают небольшие объемы данных.
- Индексированные данные в базе данных могут быть легко доступны
- Трехуровневая архитектура, которая обычно состоит из уровня представления, уровня бизнес-логики и уровня хранилища данных

Примеры OLTP



Требования к OLTP-системам

- **Соответствие ACID**
 - **Атомарность:** гарантия, что все шаги в транзакции будут успешно завершены
 - **Согласованность:** транзакция поддерживает внутреннюю согласованность БД
 - **Изолированность:** транзакция выполняется, как если бы она выполнялась одна
 - **Устойчивость:** результаты транзакции не будут потеряны в случае сбоя
- **Параллельность**
- **Масштабируемость**
- **Доступность**
- **Высокая пропускная способность и короткое время реагирования**
- **Надежность**
- **Безопасность**
- **Восстанавливаемость**

Реляционные БД для OLTP



Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Что же отличает аналитические СУБД от других?

OLAP vs. (with?) OLTP

Главная разница между OLAP и OLTP..

- ... в их названии. Analytical vs. transactional. Каждая система оптимизирована под свой тип обработки данных
- **OLAP** оптимизирована для проведения комплексного анализа данных с целью принятия более разумных решений.
 - Поддерживает бизнес-аналитику (BI), интеллектуальный анализ данных и другие приложения поддержки принятия решений.
- **OLTP** оптимизирована для обработки огромного количества транзакций

Другие ключевые различия

- **Фокус:**
 - Системы OLAP позволяют извлекать данные для сложного анализа
 - Системы OLTP, напротив, идеально подходят для выполнения простых обновлений, вставок и удалений в базах данных
- **Источник данных:**
 - База данных OLAP может поддерживать сложные запросы множества фактов данных из текущих и исторических данных. Различные базы данных OLTP могут быть источником информации.
 - OLTP, с другой стороны, использует традиционную СУБД для размещения большого объема транзакций в режиме реального времени.

Другие ключевые различия

- **Время обработки:**
 - В OLAP время отклика на порядки медленнее, чем в OLTP. Рабочие нагрузки требуют интенсивного чтения и включают огромные наборы данных.
 - Для транзакций и ответов OLTP важна каждая миллисекунда. Рабочие нагрузки включают простые операции чтения и записи через SQL, требуя меньше времени и места для хранения.
- **Доступность:**
 - Поскольку они не изменяют текущие данные, резервное копирование OLAP-систем можно выполнять реже.
 - OLTP-системы часто изменяют данные. Они требуют частого или параллельного резервного копирования для поддержания целостности данных.

Как же выбрать?

- **Вопросы:**

- Вам нужна единая платформа для получения информации о бизнесе?
- Вам нужно управлять ежедневными операциями?
- Вам нужно на постоянной основе обновлять, вставлять, удалять огромные куски информации?
- Вам нужно анализировать полученную из разнородных источников информацию и выдавать валидные бизнес-результаты?

Вопросы?



Ставим "+",
если вопросы есть

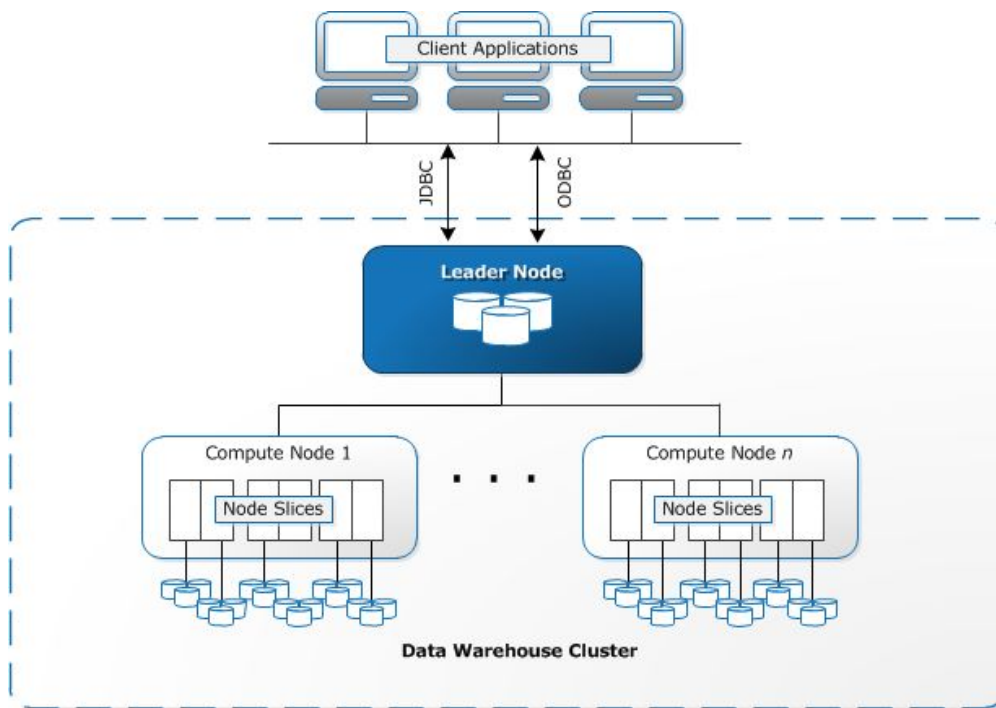


Ставим "-",
если вопросов нет



Ключевые принципы

Кластерные вычисления в основе архитектуры

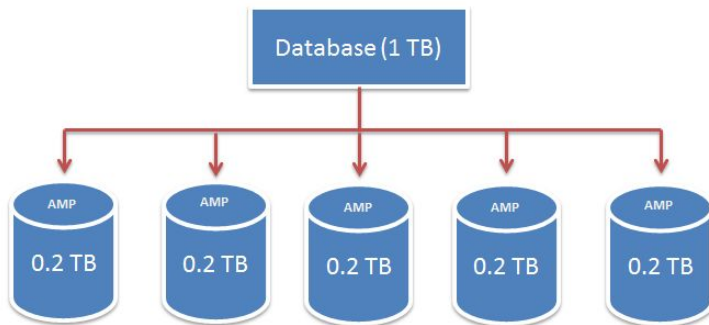


Принцип MPP

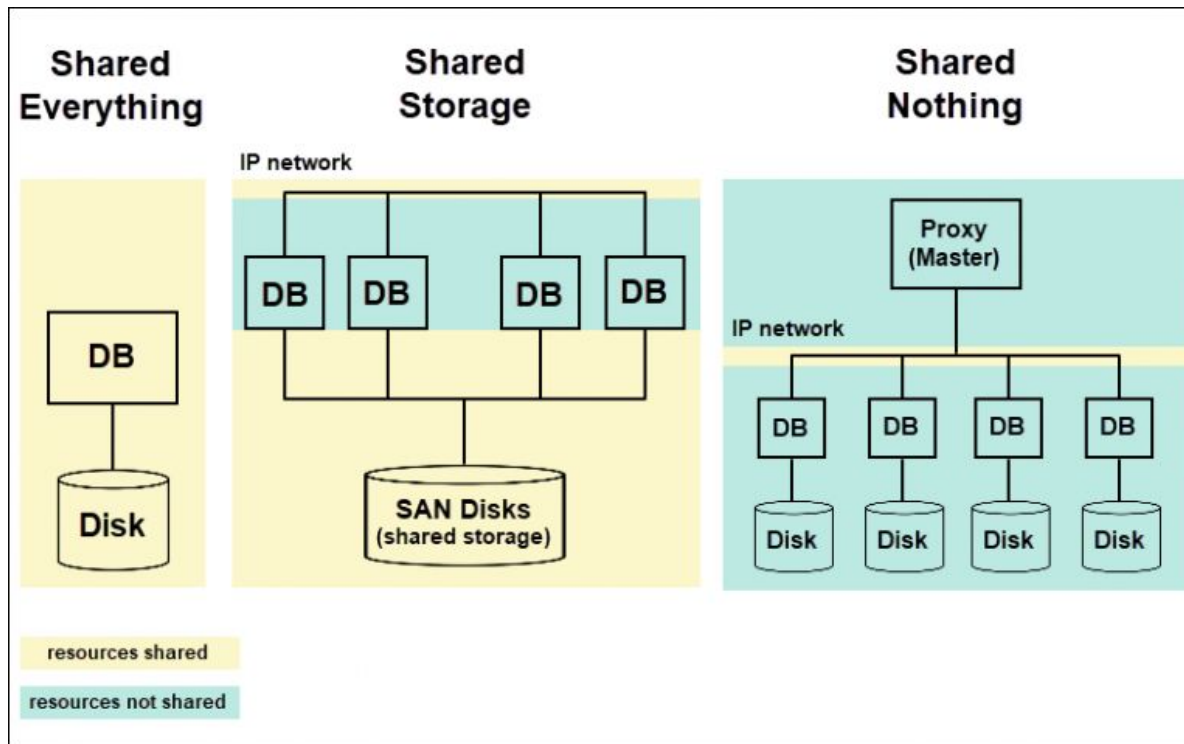
Massive Parallel Processing - массово-параллельная обработка

- Выполнение одной задачи на распределенном кластере
- Shared-nothing архитектура
- Чаще всего применяется по отношению к массивно-параллельным СУБД, которые умеют выполнять аналитические запросы сразу на всех узлах

Примеры: Greenplum, Vertica, Redshift, Teradata, Exadata



Shared-* architecture



Хранение данных по колонкам (не строкам!)

	Col A	Col B	Col C
Row 0	A0	B0	C0
Row 1	A1	B1	C1
Row 2	A2	B2	C2
Row 3	A3	B3	C3
Row 4	A4	B4	C4
Row 5	A5	B5	C5

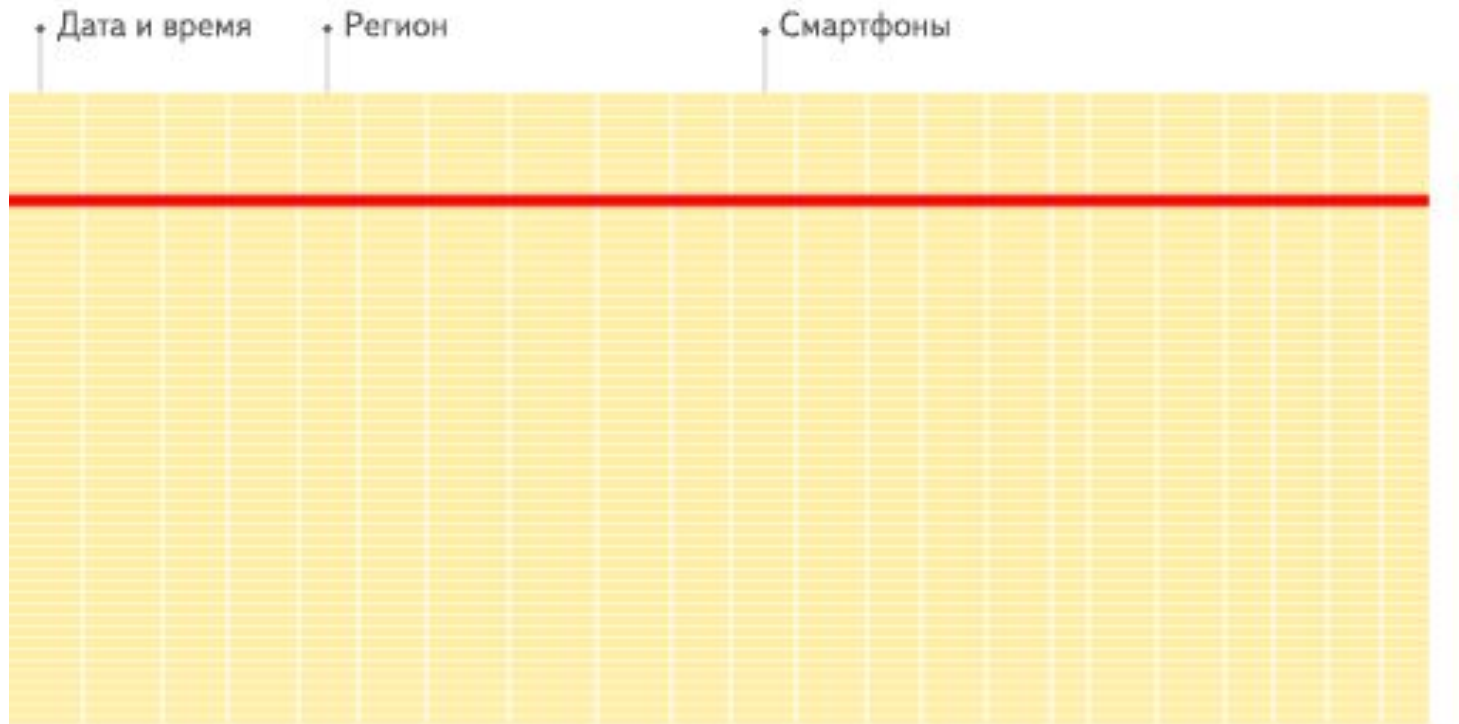


A0	B0	C0	A1	B1	C1	A2	B2	C2
A3	B3	C3	A4	B4	C4	A5	B5	C5

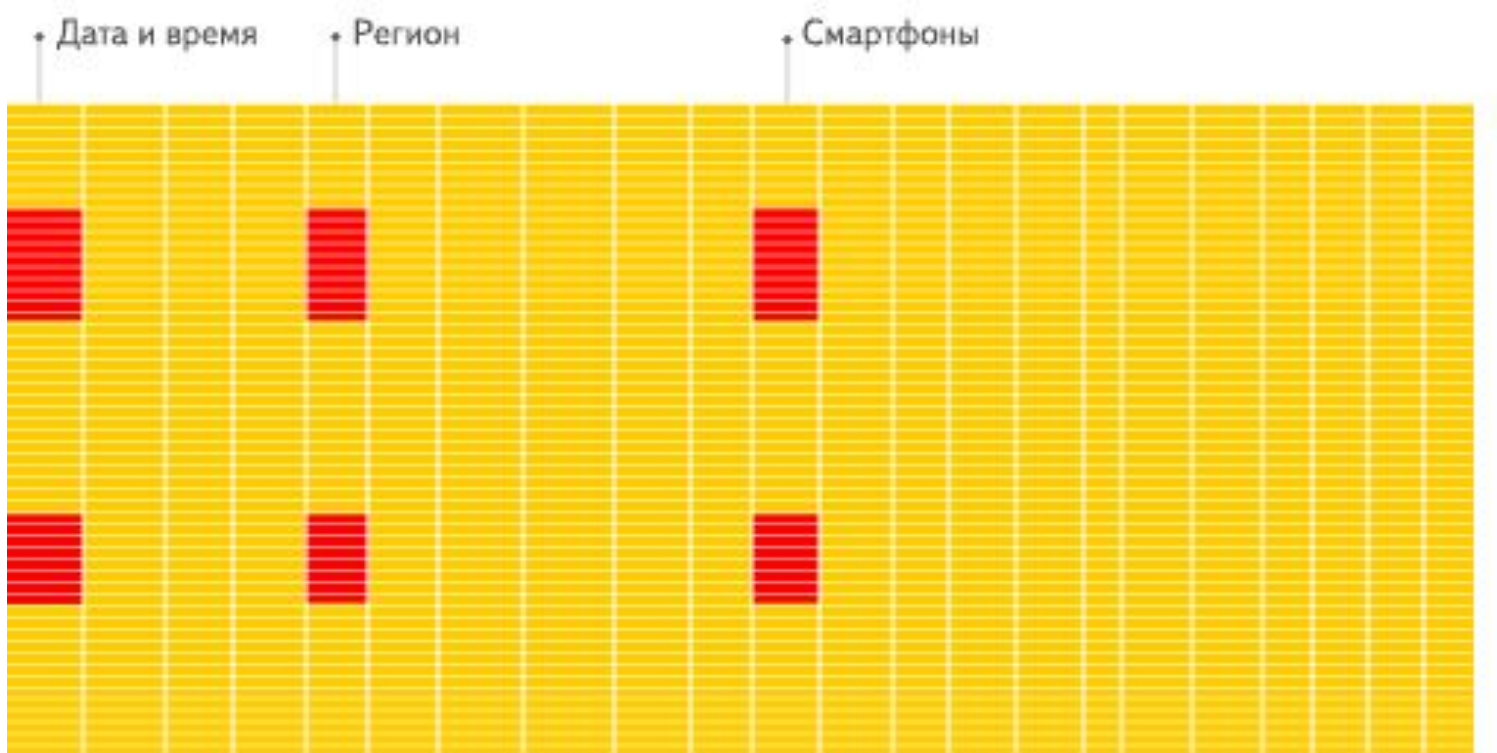


A0	A1	A2	A3	A4	A5	B0	B1	B2
B3	B4	B5	C0	C1	C2	C3	C4	C5

Наглядно - запрос к строковой СУБД



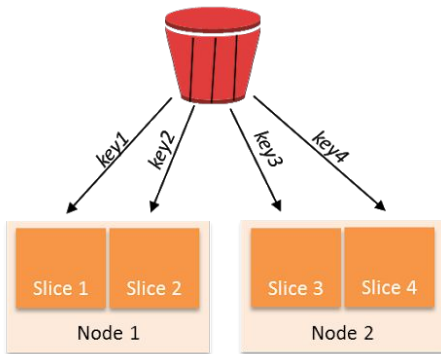
Наглядно - запрос к колоночной СУБД



Распределение записей между узлами (сегментация)

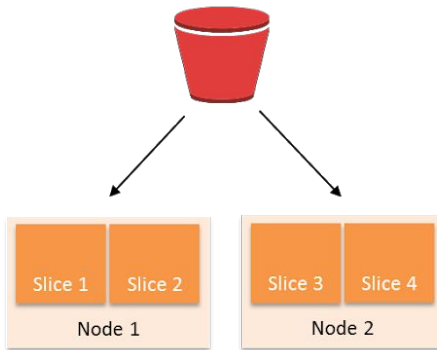
Distribution Key

key value to same location



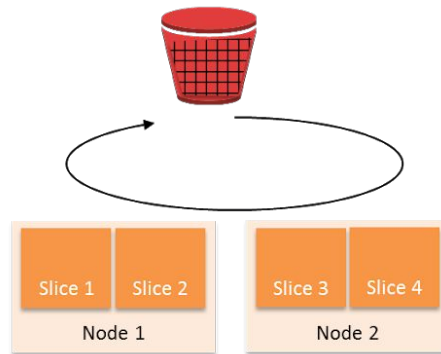
All

All data on every node



Even

Round robin distribution

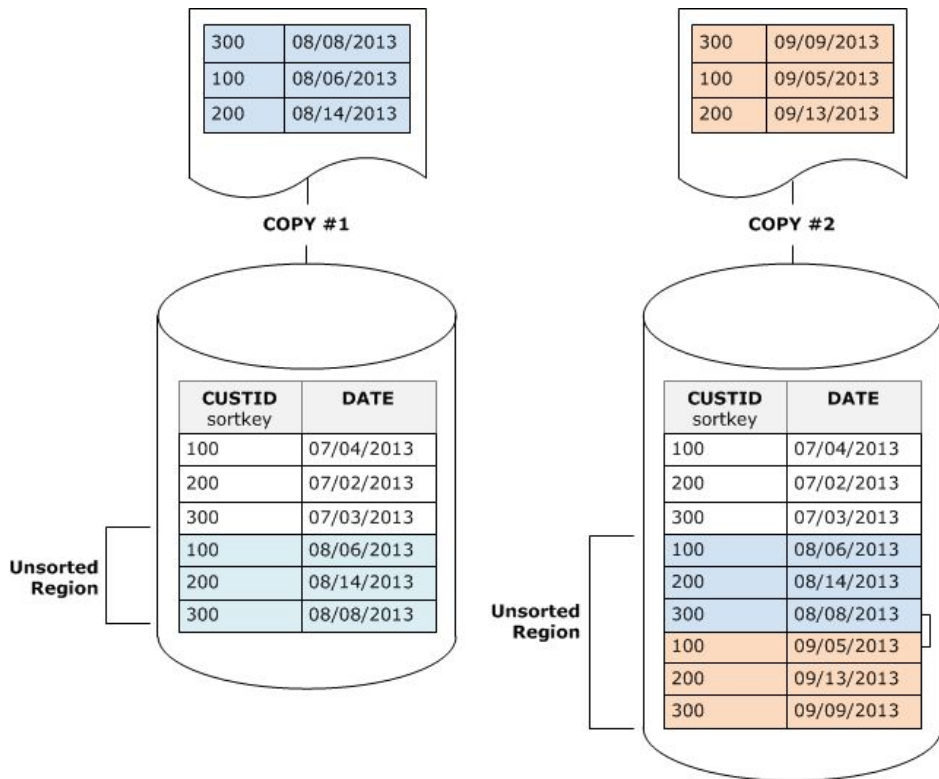


Сжатие данных (алгоритмы и кодеки)

Encoding type	Keyword in CREATE TABLE and ALTER TABLE	Data types
Raw (no compression)	RAW	All
AZ64	AZ64	SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, TIMESTAMPTZ
Byte dictionary	BYTEDICT	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Delta	DELTA	SMALLINT, INT, BIGINT, DATE, TIMESTAMP, DECIMAL
	DELTA32K	INT, BIGINT, DATE, TIMESTAMP, DECIMAL
LZO	LZO	SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER
Mostlyn	MOSTLY8	SMALLINT, INT, BIGINT, DECIMAL
	MOSTLY16	INT, BIGINT, DECIMAL
	MOSTLY32	BIGINT, DECIMAL
Run-length	RUNLENGTH	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Text	TEXT255	VARCHAR only
	TEXT32K	VARCHAR only

Zstandard	ZSTD	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER
-----------	------	-------------------------------------------------------------------------------------------------------------------------

Хранение (уже) упорядоченных данных



```
ENGINE = MergeTree
PARTITION BY toYYYYMM(date_in_poo)
ORDER BY office_id
SETTINGS index_granularity = 8192;
```

Партицирование данных (partition pruning)



Индексы и транзакции

Индексы и транзакции становятся слишком дорогими при таких объемах данных и формате хранения

Индексы и транзакции требуют синхронизации между нодами

В результате слабая/ограниченная поддержка индексов и транзакций

Вопросы?



Ставим "+",
если вопросы есть



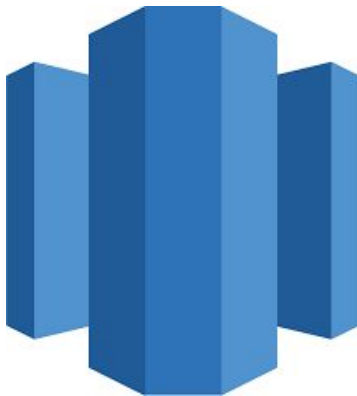
Ставим "-",
если вопросов нет



Примеры СУБД

Cloud solutions

- BigQuery (Google Cloud)
- Redshift (Amazon Cloud)
- SQL Data Warehouse (Azure Cloud)



On-prem / Hybrid

- Vertica
- Greenplum
- Teradata
- Kudu
- ClickHouse

VERTICA



GREENPLUM
DATABASE

TERADATA

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет



Best Practices

Лучшие практики MPP

- Параллельная обработка
- Сегментация
- Партицирование
- Сжатие (колоночное хранение)
- Оптимальное хранение (согласно сценариям использования)

Дорогие DELETE, UPDATE

Почему дорого удалять/изменять отдельные записи?

- Эту запись нужно найти
- При удалении/изменении одной строки приходится целиком перезаписывать большой кусок

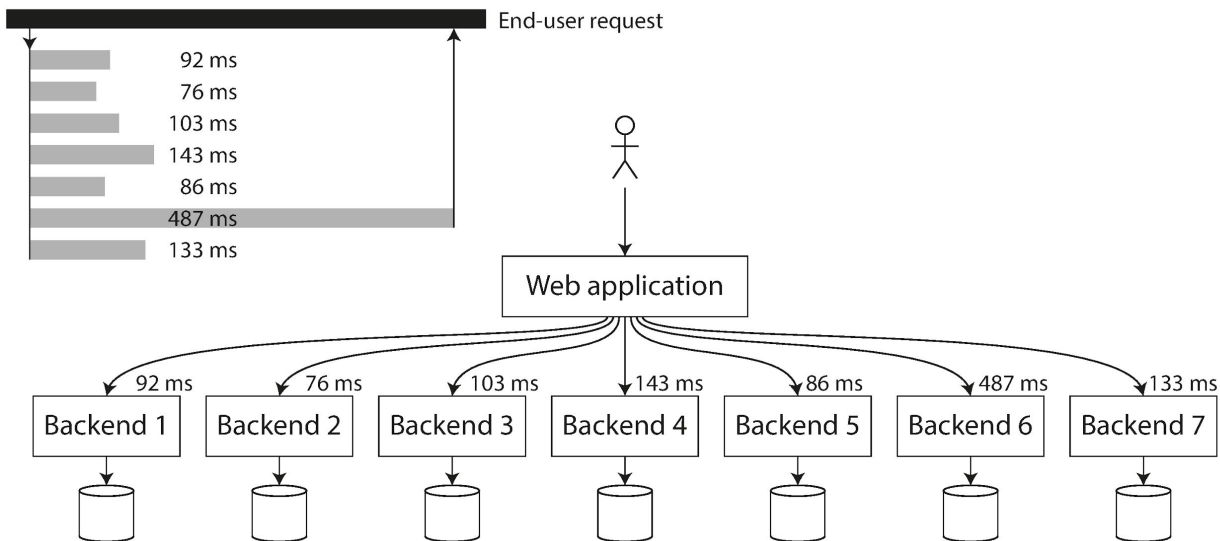
Сегментация. Зачем?

- Равномерное распределение
- Локальность операций с данными на узлах



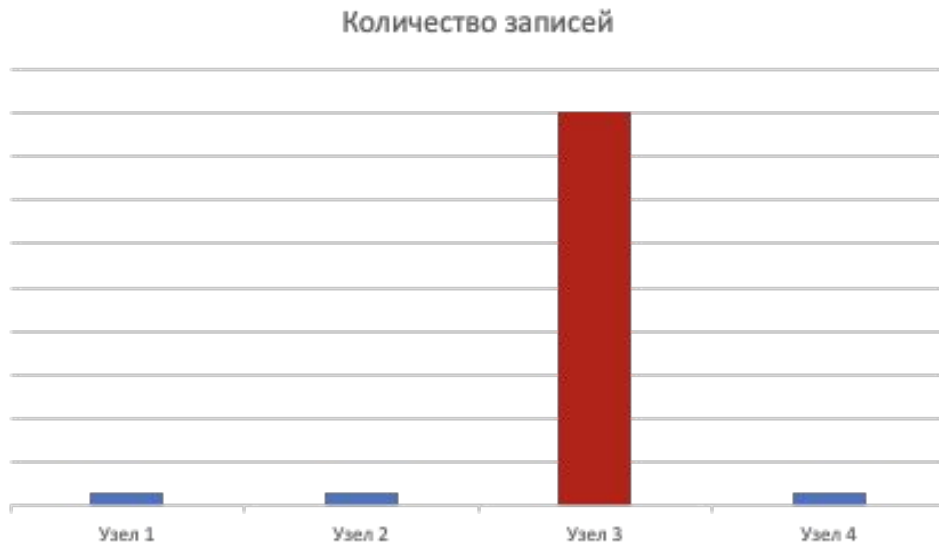
Один медленный вызов

- Когда для обслуживания запроса требуется несколько внутренних вызовов, достаточно одного медленного внутреннего вызова, чтобы замедлить весь запрос конечного пользователя.



Перекосы - потеря параллельности

- Пример: Самый крупный клиент банка
- Пример: Одна техническая УЗ на весь front-office



Как всё это конфигурировать?

```
CREATE TABLE sales (id int, date date, amt decimal(10,2))
DISTRIBUTED BY (id)
PARTITION BY RANGE (date)
( START (date '2016-01-01') INCLUSIVE
  END (date '2017-01-01') EXCLUSIVE
  EVERY (INTERVAL '1 day') );
```

You can also declare and name each partition individually. For example:

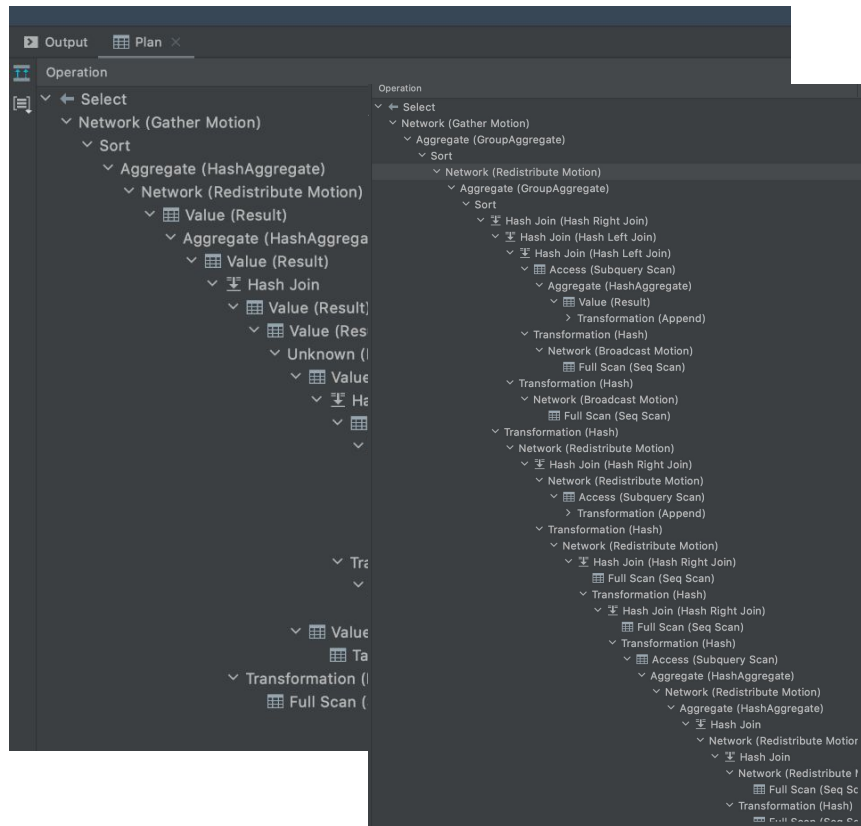
```
CREATE TABLE sales (id int, date date, amt decimal(10,2))
DISTRIBUTED BY (id)
PARTITION BY RANGE (date)
( PARTITION Jan16 START (date '2016-01-01') INCLUSIVE ,
  PARTITION Feb16 START (date '2016-02-01') INCLUSIVE ,
  PARTITION Mar16 START (date '2016-03-01') INCLUSIVE ,
  PARTITION Apr16 START (date '2016-04-01') INCLUSIVE ,
  PARTITION May16 START (date '2016-05-01') INCLUSIVE ,
  PARTITION Jun16 START (date '2016-06-01') INCLUSIVE ,
  PARTITION Jul16 START (date '2016-07-01') INCLUSIVE ,
  PARTITION Aug16 START (date '2016-08-01') INCLUSIVE ,
  PARTITION Sep16 START (date '2016-09-01') INCLUSIVE ,
  PARTITION Oct16 START (date '2016-10-01') INCLUSIVE ,
  PARTITION Nov16 START (date '2016-11-01') INCLUSIVE ,
  PARTITION Dec16 START (date '2016-12-01') INCLUSIVE
    END (date '2017-01-01') EXCLUSIVE );
```

```
id_city          integer,
start_shelf      interval,
plan_delivery    interval,
duration_200_on_shelf interval,
duration_135_on_shelf interval
)
with (appendonly = true, compresslevel = 4, compressstype = zstd)
distributed randomly;
```


EXPLAIN – анализ плана запроса

Основные операции:

- Seq Scan
- Index Scan
- Nested Loops
- Hash Join
- Sort
- Aggregate
- Unique



Остальные операции:

- Bitmap Index Scan
- Merge Join
- GroupAggregate
- HashAggregate
- Filter
- Limit
- Append
- HashSetOp
- Materialize
- CTE Scan
- SubPlan
- InitPlan
- Subquery Scan

Форматы плана запроса

```
74
75 Planning time: 160.226 ms
76 (slice0) Executor memory: 970K bytes.
77 (slice1) Executor memory: 18120K bytes avg x 72 workers, 18120K ...
78 (slice2) Executor memory: 60K bytes avg x 72 workers, 60K bytes ...
79 (slice3) Executor memory: 19592K bytes avg x 72 workers, 19592K ...
80 * (slice4) Executor memory: 10879K bytes avg x 72 workers, 193040K...
81 (slice5) Executor memory: 138K bytes avg x 72 workers, 240K byte...
82 Memory used: 589824kB
83 Memory wanted: 861599kB
84 Optimizer: Postgres query optimizer
85 Execution time: 80208.099 ms
```

QUERY PLAN

```
1 Gather Motion 72:1 (slice11; segments: 72) (cost=428761756.78..4335553...
2   -> GroupAggregate (cost=428761756.78..433555348.41 rows=887703 width...
3       Group Key: ((sh.close_dt)::date), sh.src_office_id, bo1.office_n...
4   -> Sort (cost=428761756.78..428921543.17 rows=887703 width=106...
5       Sort Key: ((sh.close_dt)::date), sh.src_office_id, bo1.off...
6   -> Redistribute Motion 72:72 (slice10; segments: 72) (c...
       _dt)::date), sh.src_office_id, b...
       (cost=305426919.37..309102006.30...
       h.close_dt)::date), sh.src_offic...
       t=305426919.37..305586705.76 row...
       : ((sh.close_dt)::date), sh.src_...
       Right Join (cost=222330841.31...
       sh Cond: (pc.rid = td.rid)
       Hash Left Join (cost=11440744...
       Hash Cond: (pc.create_dt = e...
       -> Hash Left Join (cost=11...
       Hash Cond: ((CASE WHEN...
       -> Subquery Scan on p...
       -> HashAggregat...
       Group Key:...
       -> Result...
       -> ...
```

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет



Слайд с тезисами

Подведем итоги

1. Поддержка параллельной загрузки
2. Колоночный формат хранения
3. Оптимизация данных на уровне записи
4. Дорогие DELETE, UPDATE
5. Ограниченная поддержка индексов и транзакций

Список материалов для изучения

1. [Shared Nothing Architecture Explained](#)
2. [Clickhouse vs. Greenplum. Какую MPP-базу данных выбрать? // Демо-занятие курса «Data Engineer»](#)
3. [What is an MPP Database? Intro to Massively Parallel Processing](#)
4. [Row vs Column Oriented Databases](#)

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Рефлексия

Цели вебинара

1. Провести знакомство с курсом, платформой и планом работы
 2. Разобрать примеры аналитических движков и их использование на практике
-

Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то, что узнали на вебинаре?

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Железной Алексей

*Senior Data Engineer в Wildberries
Магистратура - ФКН ВШЭ*

Руководитель курсов DWH Analyst, ClickHouse для инженеров и архитекторов БД в OTUS

Преподаватель курсов Data Engineer, DWH Analyst, PostgreSQL и пр. в OTUS