

ETL: Что это и как работает на практике

Система, которая работает без тебя

План выступления

01

Что такое ETL?

Аналогия для понимания

02

Ключевые компоненты ETL

Extract, Transform, Load — детальный разбор

03

Пакетный ETL

Мир Apache Airflow

04

Потоковый ETL

Мир Apache Kafka

05

Сравнение и выбор

Что и когда использовать?

06

Проектирование ETL

Рекомендации и ошибки

ETL — это процесс готовки данных для анализа



Extract (Извлечение)

Купить продукты на рынке — достать данные из БД, API, файлов



Transform (Преобразование)

Помыть, почистить, приготовить — очистить, объединить, посчитать



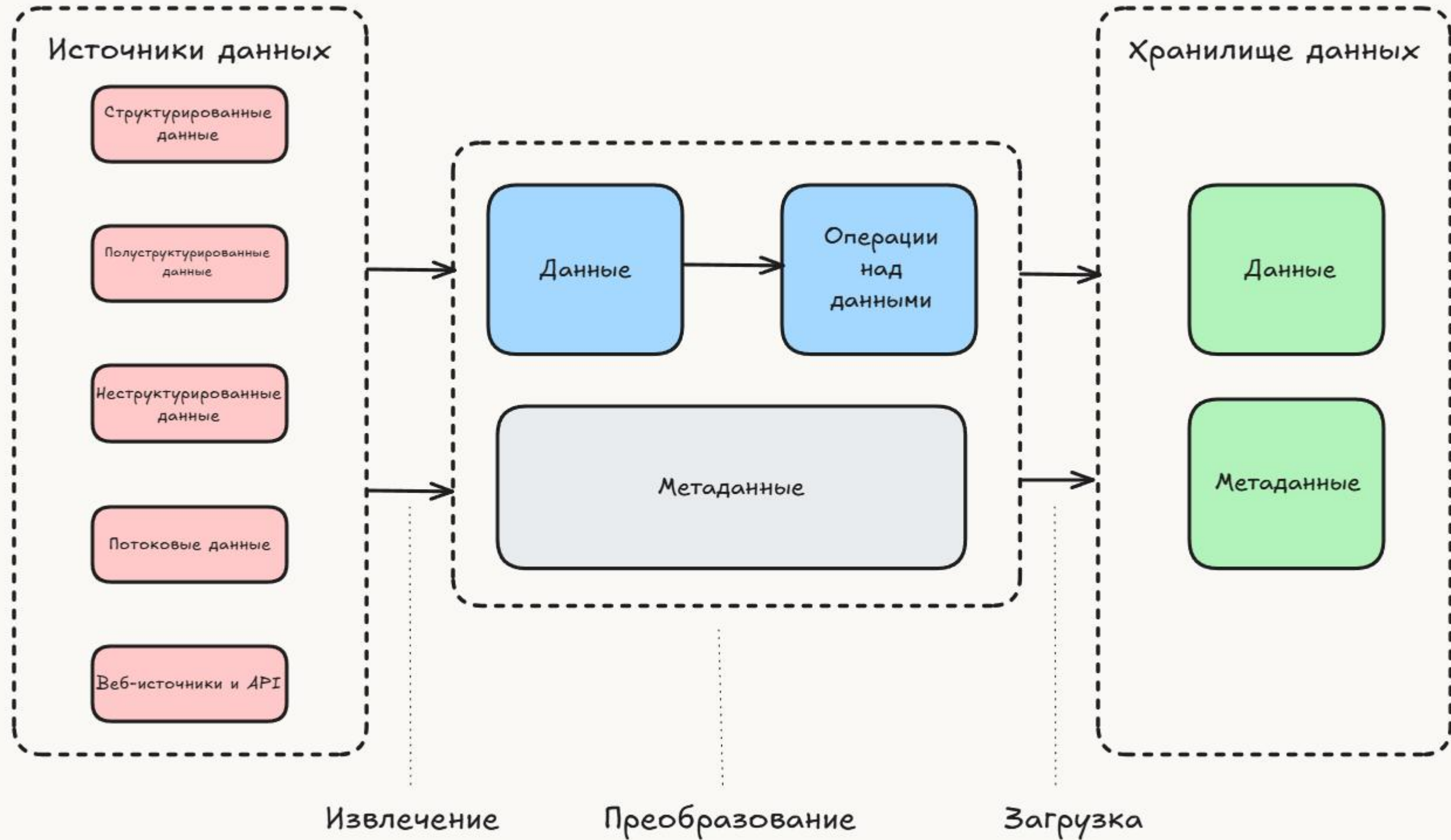
Load (Загрузка)

Подать блюдо на стол — загрузить в хранилище для анализа

Цель: Превратить разрозненные "сырые" данные в структурированную и качественную информацию для принятия решений.



Схема ETL процесса



Extract — получить данные из источников

Источники:

Базы данных: PostgreSQL, MySQL через SQL-запросы

Файлы: CSV, JSON, Parquet в облаке (S3, GCS)

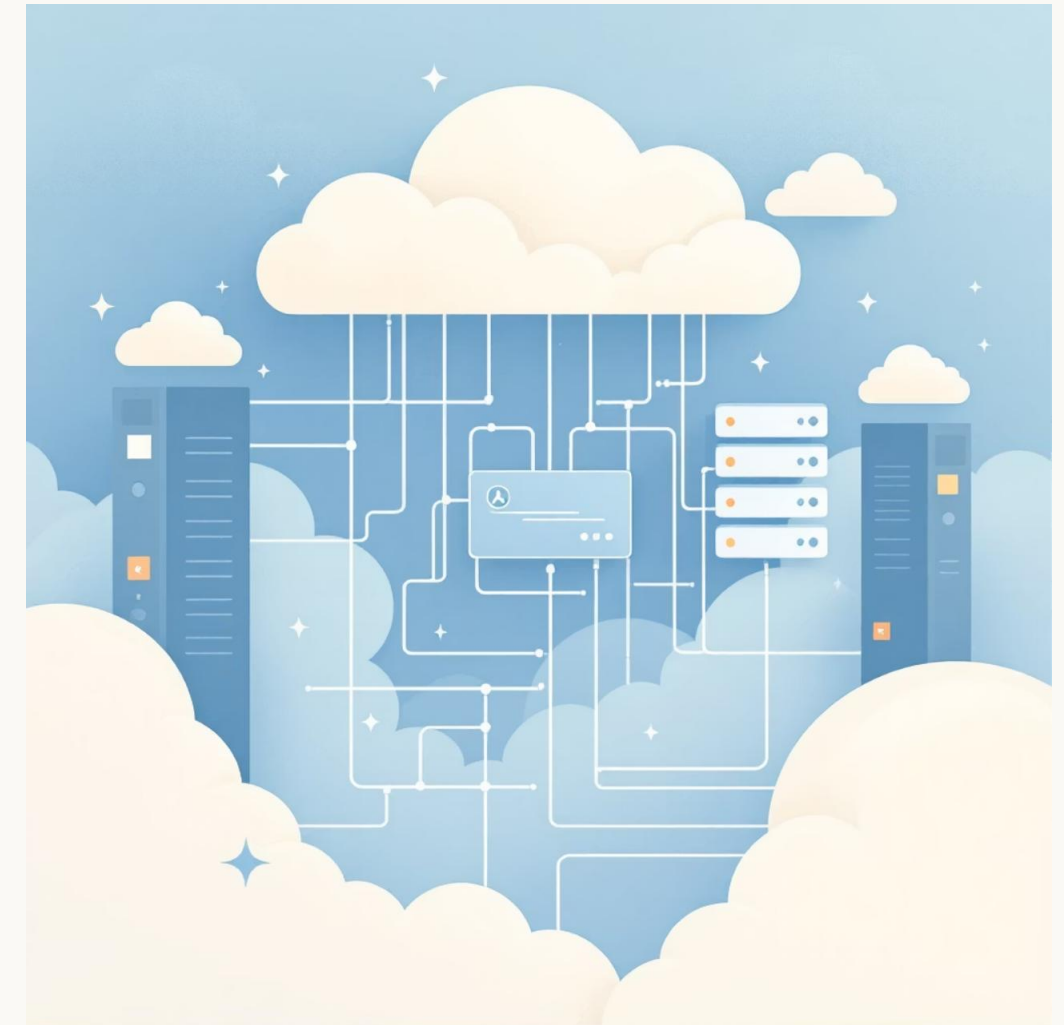
Веб-API: Данные от внешних сервисов и партнеров

Потоковые платформы: Apache Kafka

Ключевые задачи на этом этапе:

- Установить соединение с источником
- Определить объем данных
- Обеспечить надежность
- Минимизировать нагрузку

📌 Главная задача: Извлечь данные максимально эффективно





Transform — самая творческая часть работы DE

Очистка данных

Удалить дубликаты, исправить опечатки, заполнить пропуски качественными значениями

Стандартизация

Привести даты к одному формату, перевести валюты, унифицировать форматы

Обогащение

Объединить данные из разных источников — добавить к заказу информацию о клиенте

Агрегация

Посчитать итоги, суммы, средние значения, создать витрины данных

Результат: Качественные, готовые к анализу данные с бизнес-логикой.

Load — доставка результата

Целевые системы:

Озера данных (Data Lake)

Хранилища данных (Data Warehouse)

Витрины данных (Data Marts)

Стратегии загрузки:

Full Load: Просто, но медленно. Подходит для небольших справочников

Incremental Load: Эффективно. Только новые и измененные данные



Ключевые компоненты ETL-пайплайна

Что должно быть в любом надежном ETL?

✓ Источник (Source)

Откуда берём данные — четко определенные источники

✓ Преобразование (Transform)

Как обрабатываем — документированная бизнес-логика

✓ Хранилище (Sink)

Куда кладём — целевая система для анализа

+ Метаданные

Кто, когда, зачем — полная история изменений

+ Логирование

Если сломалось — быстро понять где проблема

+ Отказоустойчивость

Возможность безопасного перезапуска процессов

+ Повторяемость

Один вход → один выход. Детерминированность результатов

Пакетная обработка — работа по расписанию

Аналогия: Грузовик, который перевозит большой объем товара раз в день

Принцип работы:

Обработка данных крупными порциями по заданному расписанию — раз в час, день или неделю.

Плюсы:

- Высокая эффективность для больших объемов
- Проще в отладке и управлении
- Меньше ресурсов на инфраструктуру

Минусы:

- Данные не обновляются в реальном времени
- Высокая задержка (latency)

💡 **Инструмент: Apache Airflow.**
Создаете DAG, который запускается по таймеру и обрабатывает данные за прошедший период.





Потоковая обработка — данные в реальном времени

Аналогия: Конвейерная лента, обрабатывающая предметы поштучно и непрерывно

📌 **Инструмент: Apache Kafka**
Центральная "артерия" для потоковых данных. События поступают и сразу обрабатываются.

Принцип работы:

Данные обрабатываются по мере их появления, почти без задержки.

Плюсы:

- Минимальная задержка обработки
- Возможность мгновенного реагирования
- Обработка событий в реальном времени

Минусы:

- Сложнее в проектировании и отладке
- Требуется больше ресурсов на обеспечение надежности

Batch vs. Streaming — два инструмента для разных задач

Критерий	Batch (Пакетная)	Streaming (Потоковая)
Задержка	Часы/дни	Секунды/миллисекунды
Объем данных	Большие партии	Непрерывный поток мелких событий
Сложность	Проще в реализации	Требует экспертизы
Стоимость	Ниже	Выше

Use Cases для Batch

Ежедневные отчеты, обучение ML-моделей, исторический анализ

Use Cases для Streaming

Мониторинг мошенничества, онлайн-рекомендации, алерты

Ключевой вывод: Нет "лучшего" подхода. Есть бизнес-задача, и под неё выбирается адекватный инструмент. Часто системы используют оба подхода одновременно (Lambda Architecture).

ETL: важные характеристики

⚠️ Обработка ошибок

Не падайте молча!

🔄 Идемпотентность

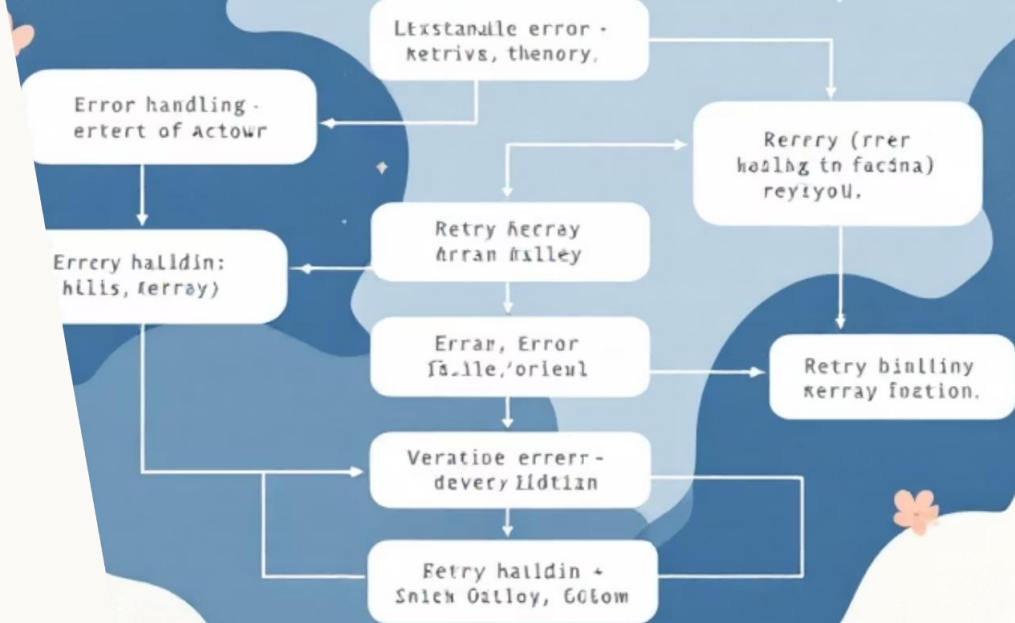
Гарантия от дубликатов - одинаковый результат при повторном запуске

📊 Мониторинг

Настройте комплексную систему отслеживания состояния системы

✓ Валидация данных

Проверяйте данные до и после преобразований



Источники данных: откуда всё начинается

PostgreSQL

Основная база с заказами, товарами и транзакциями

CRM API

Данные о клиентах через REST API интеграцию

Kafka

Поток кликов и событий с сайта в реальном времени





Batch ETL: ночная обработка данных

01

Extract (Извлечение)

SQL-запросы к PostgreSQL и вызовы CRM API для получения данных за день

02

Transform (Преобразование)

Очистка данных от дубликатов, агрегация показателей по товарам и категориям

03

Load (Загрузка)

Сохранение в аналитическое хранилище для построения отчётов

🕒 Запуск каждый день в **02:00** через Airflow для создания витрины данных



Streaming ETL: данные в реальном времени

01

Extract (Извлечение)

Чтение из Kafka: постоянное получение событий кликов и действий пользователей

02

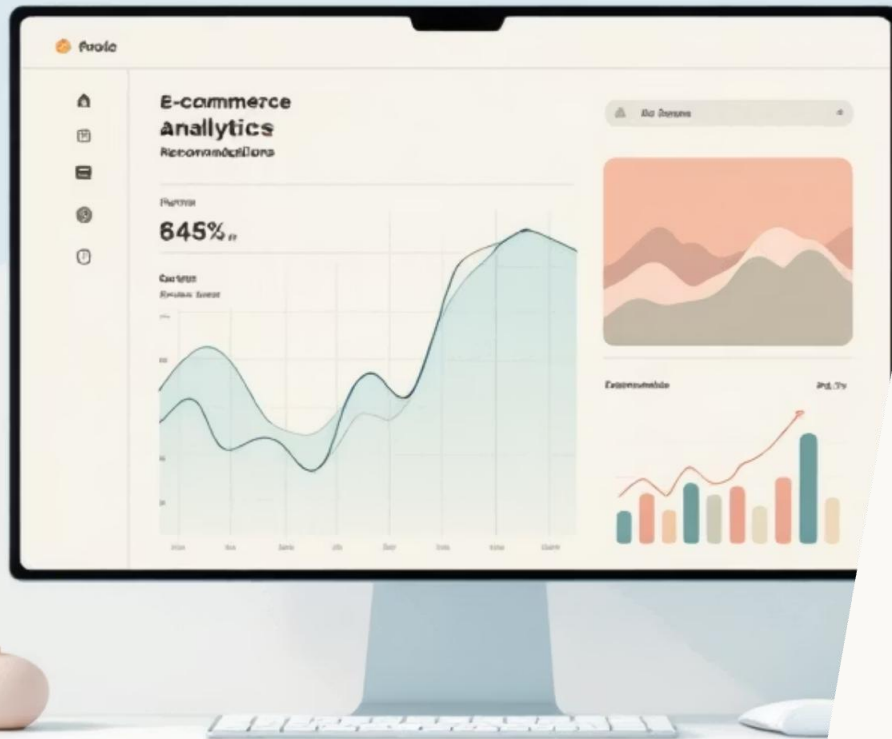
Transform (Преобразование)

Подсчёт популярности товаров и формирование рекомендаций

Load (Загрузка)

Сохранение в Redis для быстрого доступа с сайта

Конечные потребители данных



☑ Дашборд в Superset

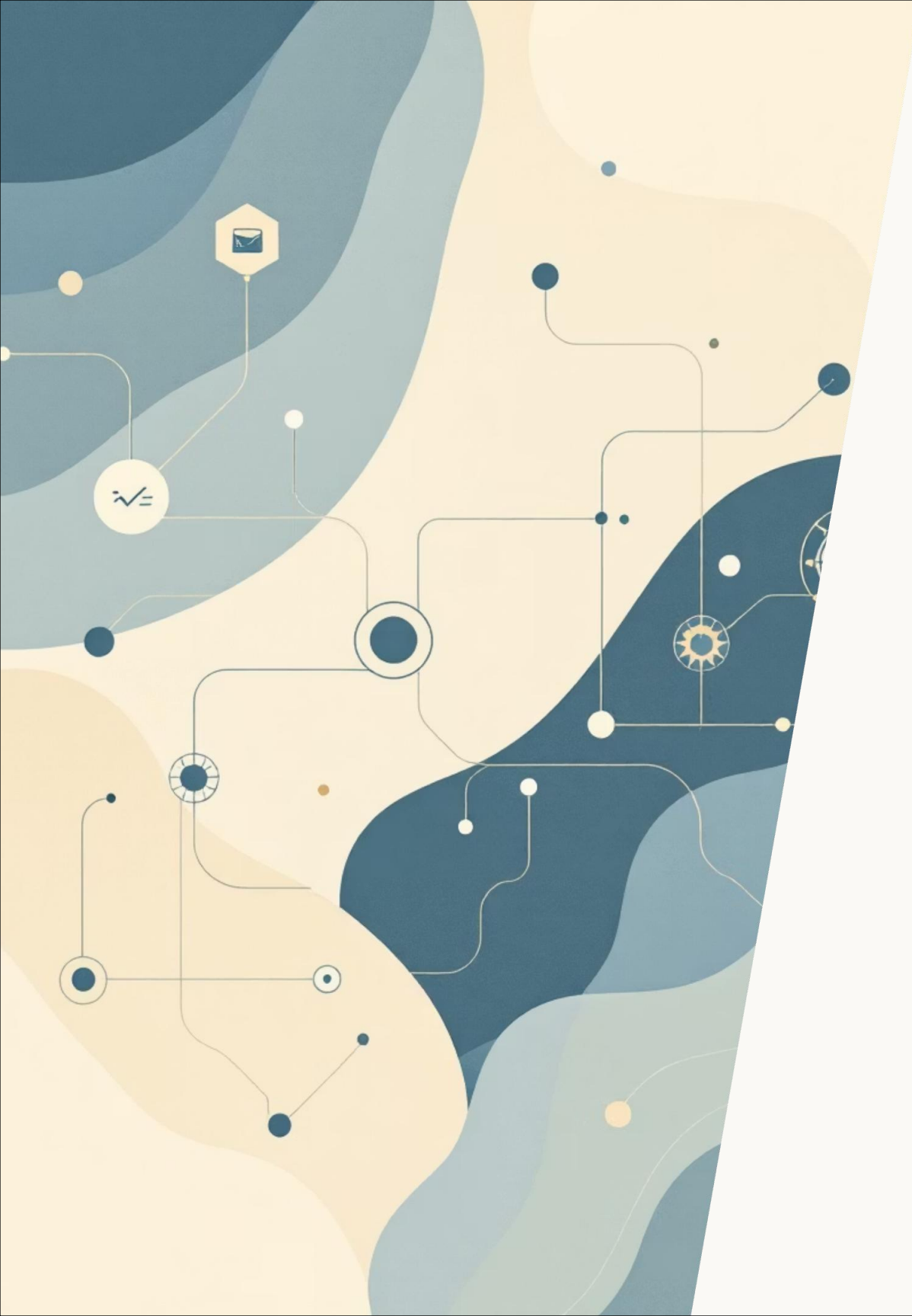
Отчётность по продажам для менеджеров и аналитиков

- Выручка по регионам
- LTV клиентов
- Конверсия корзины

🌐 Онлайн-рекомендации

Интеграция с сайтом для показа актуальных трендов

- "Тренды недели"
- Персонализация



Ключевые идеи, которые стоит запомнить

ETL/ELT — фундамент Data Engineering

Процесс превращения "сырых" данных в качественную информацию для бизнеса

Batch и Streaming — инструменты для разных задач

Не конкуренты, а дополняющие решения. Выбор зависит от требований к задержке данных

Надежность определяется "обязкой"

Обработка ошибок, мониторинг и идиempотентность важнее самого кода

Концепции первичны, инструменты вторичны. Понимание теории ETL позволит освоить любые технологии



Thank You

Спасибо за внимание!

Вопросы?

Подписывайтесь на канал и группы

Продолжайте изучать концепции и применять их
на практике для создания надежных решений

- Telegram-канал: t.me/marat_notes
- Обучающие видео: <https://vkvideo.ru/@club231048746>, https://www.youtube.com/@marat_notes
- Репозиторий: https://github.com/MaratNotes/marat_notes