СЛЁРМ Обучение для инженеров и технических лидеров

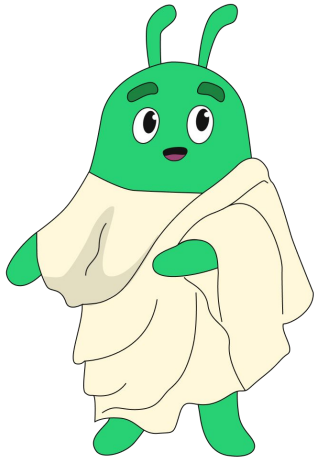
Дата-инженер

Процессы ETL и ELT

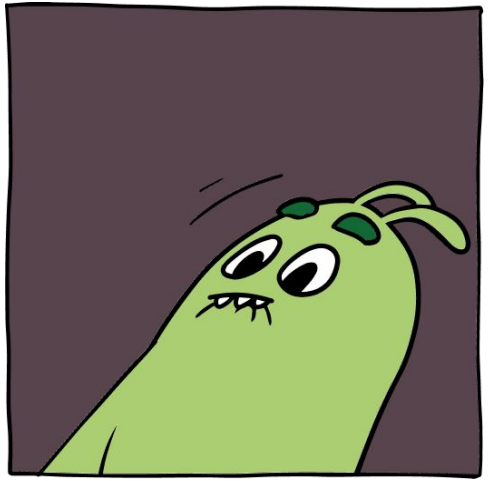
Николай Марков

Цели урока. Что вы узнаете:

1. Какие задачи решают инструменты ETL
2. В чем суть каждой из стадий — “E”, “T” и “L”
3. Какая разница между ETL и ELT и когда применять одно или другое

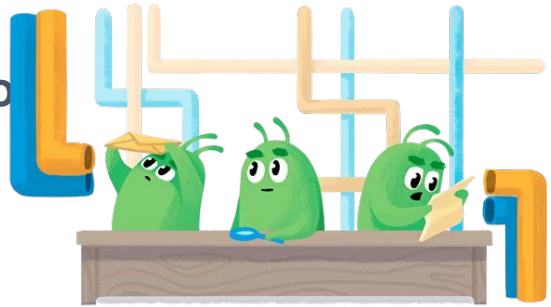


Зачем нужен ETL



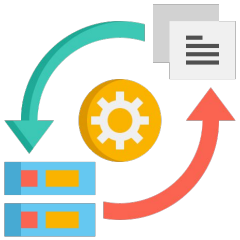
Перекладывание данных

- У организации есть **источники данных**: например, логи поведения пользователей на сайте
- Эти данные нужно **выгрузить и положить в хранилище** неструктурированных (**Data Lake**) или структурированных (**Data Warehouse**) данных
- Также может потребоваться переложить данные **из неструктурированного в структурированно**



Перекладывание данных

Извлекаем (Extract)



Преобразуем (Transform)

Загружаем (Load)

Что мы хотим по итогу

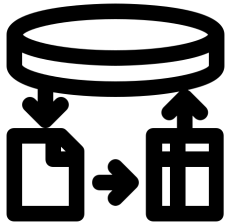
Какие цели мы преследуем:

1. **Достать** данные из источника в виде **пакетов/файлов** или **потока**
2. **Отфильтровать** данные по необходимым признакам согласно тому, что требуется для анализа
3. **Стандартизировать** и привести к нужному виду для заливки дальше: например, в формате **CSV, Parquet** или **Avro**
4. **Залить** данные в хранилище

ETL

По сути, почти любой процесс перекладывания данных с места на место можно «обозвать» ETL

Однако чаще всего имеется в виду именно **заливка данных из источников в хранилище**

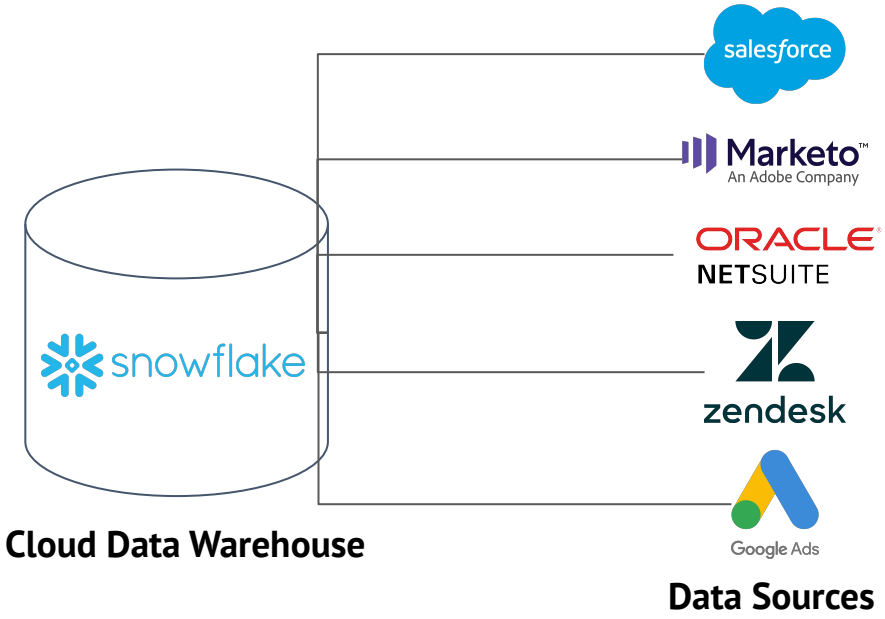


Зачем мы все это делаем

- Бизнесу необходимо **быстро** получать **релевантные инсайты** на основе своих данных в процессе **Master Data Management**
- Для этого должен существовать **доверенный источник** данных, на основе которого можно делать аналитику — **SSOT (Single Source of Truth)**
- В некоторых случаях дополнительно используется термин **Golden Record** — это эталонный набор данных, например, о клиентах

Проще говоря, ETL забирает данные и превращает их в нужный для бизнес-аналитики вид

Reverse ETL



Окей, мы загрузили данные в хранилище.

Как нам теперь поработать с ними в наших корпоративных аналитических системах?

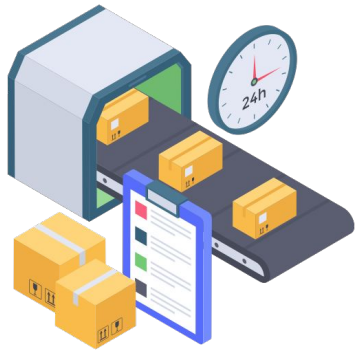
Суть этапов Extract, Transform и Load

E — Extract

Этап извлечения данных из первичных источников

Пакетная обработка (batching)

Исходные данные извлекаются **большими порциями** из источника данных в целевую систему **по расписанию** (через запланированные промежутки времени)

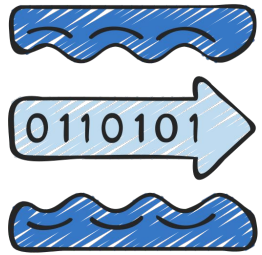


E — Extract

Этап извлечения данных из первичных источников

Потоковая обработка (streaming)

- Данные как непрерывно извлекаются из источника данных в режиме реального времени, так и поступают в источник данных
- Один из вариантов стриминга — потоковая обработка маленьких порций данных **(microbatching):**
- Этот способ предназначен для того, чтобы в режиме реального времени в процессе передачи данных перед загрузкой в хранилище осуществлять над ними преобразования



T — Transform

Этап преобразования извлеченных данных

Очистка и валидация данных

- исправление любых ошибок и заполнение отсутствующих значений
- удаление дубликатов
- логирование невалидных записей, полученных из источника

Есть даже такая позиция — Data Janitor

T — Transform

Этап преобразования извлеченных данных

Подготовка данных

- нормализация
- обогащение
- кодирование, анонимизация и обезличивание
- объединение данных из разных источников
- структурирование, преобразование одного формата в другой
- генерация бизнес-идентификаторов

T — Transform

Этап преобразования извлеченных данных

Оптимизация данных для потребителей

- сортировка
- фильтрация
- агрегирование

L — Load

Этап загрузки данных в хранилище

- **Первичная загрузка** данных в хранилище
- **Инкрементальная загрузка** — это периодическое (по расписанию или событию) добавление новых данных или модификация существующих
  в соответствии с изменениями в первичных источниках данных
- **Полное обновление** данных в хранилище — это удаление содержимого одной или нескольких таблиц и загрузка свежих данных

На этом этапе имеет смысл посчитать еще и метрики качества данных — Data Quality

Резюмируем

С появлением инструментов для потоковой обработки данных всё чаще используется для обработки данных о событиях в реальном времени

ETL позволяет получать из источников данные для анализа и загружать их в хранилище

Исторически использовался для реализации пакетной обработки данных в больших масштабах

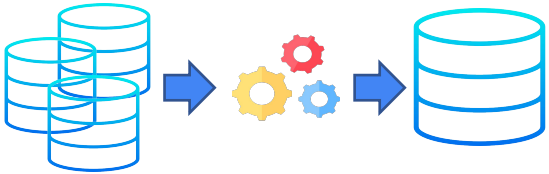
Применимость
и различия ETL и ELT

Процесс ETL и ELT

Процессы для перемещения данных из исходной системы в целевую можно разделить на две группы. Они различаются порядком выполнения операций:

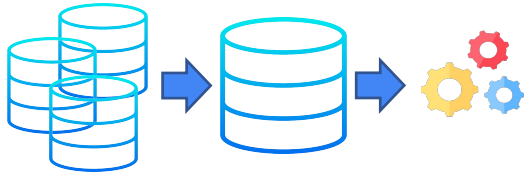
ETL (extract, transform, load)

Процесс извлечения, преобразования и загрузки данных



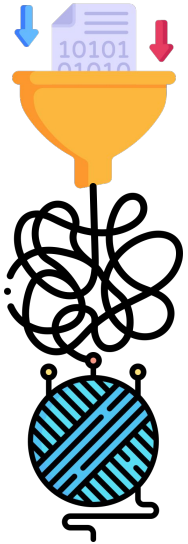
ELT (extract, load, transform)

Процесс извлечения, загрузки и преобразования данных



Как ELT взаимодействует с данными

- В процессе ELT извлеченные данные **сначала загружаются в целевую систему**
- Далее **в целевой системе** осуществляются преобразования над данными
- После этого данные уже находятся в **структурированной форме**, пригодной для аналитики
- Целевым хранилищем может быть как **Data Lake,** так и **Data Warehouse**



ELT

Необработанные данные доставляются **непосредственно в целевую систему,**
а не в промежуточную среду, что **сокращает цикл** между извлечением
и доставкой данных

В сочетании с озером данных
в целевой системе позволяет получать **свежие необработанные данные**
(как только они становятся доступными
в источнике данных)

Получил распространение
в результате развития облачных платформ

ETL

Позволяет **удалить чувствительные записи** из данных еще перед загрузкой их в целевую систему

Позволяет **изменять структуру** еще на этапе заливки данных

Появился раньше,
чем ELT, существует много готовых инструментов

Итоги. О чем поговорили:

1. ETL и ELT применяются для перекладывания данных, но они различаются методикой

2. **E - extract** – извлечение данных
   T - transform – преобразование извлеченных данных
   L - load – загрузка данных

3. ELT кажется неочевидным, но на деле имеет ряд серьезных преимуществ перед ETL

Спасибо
за внимание!