Data Engineer

Лучшие практики по приготовлению данных. Загрузка, обработка, организация хранения и доступа к данным с использованием современных инструментов

Длительность курса: 122 академических часа

1 Data Engineer: задачи, инструменты, умения

1 **Инженер Данных. Задачи, навыки, инструменты, потребности на рынке.**

- Кто этот специалист и зачем?
- Какая ценность для бизнеса?
- Потребность на рынке. Навыки
- Дальнейшее развитие
- Data Driven Decisions

- Какие задачи решает?
- Инструменты для решения основных задач
- Введение в большие данные: где используют, экосистема и история развития фреймворков

Домашние задания

1 Домашнее задание: анализ рынка Инженер Данных: РФ, USA, EU

Топ потребностей бизнеса, ключевые технологии и умения
Понять для себя, где и чем бы вам хотелось заниматься
Фиксация целей на процесс обучения

2	**Эволюция подходов работы с данными. Базовые принципы и понятия.**	- CAP theorem, Distributed Computing, MPP (Massive Parallel Processing). - Эволюция аналитических хранилищ данных - SQL + Not Only SQL. - Batch + Stream - Lambda, Kappa Домашние задания 1 Строим свою архитектуру Цель: В данном домашнем задании от участников курса ожидается небольшой архитектурный документ (желательно - не более 3-х страниц). Выберите интересный сервис (Twitter / Uber / Linkedin / ваша организация или собственный проект) и разработайте для него архитектуру аналитического хранилища данных. Опишите возможные требования к хранилищу, источники и архитектуру хранения. Приведите пример бизнес-кейса, который можно решить с помощью выбранной вами архитектуры. В решении обязательно должна присутствовать архитектурная схема вашего решения, которая должна объяснять откуда к вам поступают данные, как вы планируете их хранить и как вы планируете их отдавать для решения бизнес-кейса. Описание архитектуры - это стандартное задание на архитектурных собеседованиях для Data Engineer. Выполнив данное задание, вы сможете в будущем воспользоваться этими знаниями для того, чтобы качественно и продуманно создавать дизайн для аналитического хранилища данных или понимать как ваше data-driven приложение ложится в экосистему вашей организации. Для рисования схемы советуем использовать бесплатный сервис draw.io Разместите ответ в виде документа Google Docs и поделитесь ссылкой.
3	**Облачные платформы. GCP, AWS, Azure.**	Облака: Amazon Kinesis, Google Cloud Pub-Sub, Google Dataflow. Cloud functions Домашние задания 1Создание пайплайна обработки данных на Google Cloud Platform Цель: В этом домашнем задании мы создадим простейший пайплайн обработки данных 1) Развернем виртуальную машину, которая будет играть роль продуктового сервиса, генерирующего данные 2) Зальем данные в хранилище GCS 3) Загрузим эти данные в BigQuery для возможности анализа на SQL Ссылка на инструкцию к домашнему заданию https://docs.google.com/document/d/1OENWwIIuHYC0tMmxeySGBIFYtlSiflbTXXRKSiNSuis/edit?usp=sharing
4	**Дистрибутивы Cloudera и Hortonworks**	- Кто такие Cloudera и HortonWorks и что за экосистемы они строят - Как собрать кластер Hadoop на основе популярных дистрибутивов CDH и HDP

2 Загрузка и форматы данных (Data Ingestion)

1	**Распределенные файловые системы**	- Принципы работы распределенных файловых систем - Структура кластера HDFS - Тонкости настройки HDFS - конфигурация, защита, обеспечение отказоустойчивости
2	**Инструменты выгрузки данных из сторонних систем - 1 часть**	- Типы систем-источников. Структурированные, полу- и неструктурированные данные. Логи, выгрузки из АС, Clickstream - Инструменты для извлечения и загрузки данных - Flume, Sqoop, StreamSets, Fluentd, Debezium, logstash - Практические примеры загрузки данных из сервисных баз данных

3 **Инструменты выгрузки данных из сторонних систем - 2 часть**

- Типы систем-источников. Структурированные, полу- и неструктурированные данные. Логи, выгрузки из АС, Clickstream
- Инструменты для извлечения и загрузки данных - Flume, Sqoop, StreamSets, Fluentd
- Практические примеры загрузки данных из сервисных баз данных

Домашние задания

1 Создать снепшот аналитической таблицы из операционного хранилища

Цель: Зачастую в Data Lake не требуется хранить всех сырых данных, например в ситуации когда наиболее интересными для пользователя являются какие-либо аналитические выгрузки. В нашем домашнем задании мы напишем пример такого приложения, которое позволяет писать в Data Lake текущий аналитический срез.

1. Скопируйте на свою рабочую станцию (ПК или Google Cloud) данный репозиторий: https://github.com/renardeinside/vogel
2. Перейдите в папку проекта (vogel) и запустите все необходимые инстансы командой:
docker-compose up --build
3. Сгенерируйте токен для доступа к Jupyter Notebook (выполните эту команду в проектной папке):
docker-compose -f docker-compose.yml exec jupyter-local jupyter notebook list
4. Перейдите в браузер и введите в нем полученную ссылку с токеном.
5. В Jupyter Notebook перейдите в папку work, в ней откройте ноутбук postgres_snapshot.
6. Запустите в нем все шаги до шага Read and join the data.
7. В переменную query запишите select-запрос, который выводит следующие данные (доступные таблицы - customers, products, orders):

id - id пользователя из таблицы customers
first_name - first_name пользователя из таблицы customers
last_name - last_name пользователя из таблицы customers
total_weight - суммарный размер всех продуктов данного пользователя (поле weight в таблице products)
load_dttm - дата/время выполнения команды загрузки

Выполните данную ячейку командой Shift + Enter.

Дополнительные ограничения запроса - необходимо выграть только тех пользователей, у которых id<=1005.
Информацию по структурам таблиц вы можете прочесть в README.md файле.

8. Проверьте что были записаны верные данные в следующей ячейке.
9. В комментарии к ДЗ пришлите ваш форк данного репозитория с верной sql-командой.

---

4 **Форматы данных и их особенности**

- Назначение row-based и column-based форматов
- Обзор наиболее распространенных форматов: Avro, Parquet, ORC

3 Построение пайплайнов / Очереди сообщений / Хранилища данных

1 **Очереди сообщений**

- Kafka, RabbitMQ
- Потоковая обработка (виды обработки, описание Producer–consumer problem, пример архитектурного решения через Kafka, RabbitMQ, NATS)
- Google Dataflow paper (Event time vs processing time и так далее).
- Паттерны stream processing Joins, enricher, router. Event-sourcing.

Домашние задания

1 Архитектурный анализ применимости очереди сообщений для конкретного кейса

Цель: В этом ДЗ мы оцениваем архитектурные решения (очередь сообщений/файловый обмен/RPC) для ваших кейсов, пробуем рассуждать о архитектурных критериях, ключевых требованиях

1. Предложите задачу построения взаимодействия двух приложений близких к вам (примеры - слайды 18, 20)
2. Используя критерии на слайдах 15-16 (или предложите свои) проведите анализ вариантов взаимодействия и сделайте обоснованную рекомендацию

2 Анализ стримингового приложения по модели Dataflow

Цель: В этом ДЗ мы изучаем модель Dataflow и рассуждаем о простом стриминг приложении в концепциях Dataflow

1. (слайд 57) Рассмотрим стриминг приложение которое считает остаток на счете дебетовой карты:
• Получает события вида “timestamp, account, amount”, где amount может быть положительным или отрицательным
• Максимальное время задержки доставки событий: 2 секунды
• Если на счете недостаточно средств – приложение выдает алерт и прежний остаток, если достаточно – обновленный остаток
2. Прочитайте Dataflow paper (приложено в материалах) и развернуто ответьте на четыре вопроса описывающее это стриминговое приложение:
• Что считаем? (What?) Что считает приложение? Описание трансформации над событием
• Где по времени события? (Where?) Где во времени (над какими событиями на линии времени) происходят трансформации? Описание используемых окон
• Когда по времени обработки? (When?) Когда по времени обработки материализовать результат трансформации? Описание водяных знаков и триггеров
• Как связаны? (How?) Как связаны последующие результаты трансформации с предыдущими? Описание агрегации (дополнительное задание: опишите как бы вы поступили в случае поздней доставки данных, если задержка более двух секунд)

---

2 **DWH. Хранилища данных - 1 часть**

- Семейство MPP баз - назначение и особенности
- Логический и физический дизайн
- Vertica

3	**DWH. Хранилища данных - 2 часть**	- Семейство MPP баз - назначение и особенности - Логический и физический дизайн - Google BigQuery Домашние задания 1 Домашнее задание: проектирование витрины в Vertica (BigQuery). Цель: Спроектировать схему данных + Построить витрину Использовать Vertica (Docker) или BigQuery • Датасет: Захват данных из divolte (или GCP Public Datasets) Definition of Done: • DDL объектов • DML шагов преобразовании • Опционально: Тестирование на наличие ошибок в данных
4	**Хранилища NoSQL. Назначение и особенности.**	- NoSQL Databases. HBase, Cassandra, Elasticsearch, Aerospike - Key-value - Cache
5	**SQL-доступ к данным. Apache Hive.**	Домашние задания 1 HiveQL Цель: Практика с Hive на CDH Развернуть дистрибутив CDH Самостоятельно проделать манипуляции с Hive с приложенными скриптами
6	**Confluent Platform**	- Apache Kafka & Confluent platform - Schema registry. Данные с фиксированной схемой. - KStreams. Фреймворк для потоковой обработки. - KSQL. SQL на потоках данных.
7	**Elasticsearch**	- Знакомство с компонентами ELK-стэка - Классы задач, для которых подходит ELK

4 Процессинг и доступ к данным

1	**Apache Spark - 1 часть**	- Spark - что это и зачем он нужен - API - RDD, Dataset, Dataframe, операции над распределенными коллекциями - Процесс вычисления в Spark - task, stage, оптимизатор запросов
2	**Apache Spark - 2 часть**	- Spark - что это и зачем он нужен - API - RDD, Dataset, Dataframe, операции над распределенными коллекциями - Процесс вычисления в Spark - task, stage, оптимизатор запросов
3	**Spark Streaming**	- Micro-batch обработка данных - Классический Spark Streaming - Structured Streaming - Continuous processing
4	**Доступ к данным, ноутбуки. Explore and visualize.**	- Инструменты интерактивной аналитики - Google Cloud Datalab - Jupyter - интеграция с Apache Spark
5	**Обучение моделей. ML.**	Пример построения модели Домашние задания 1 Задание: обучаем собственную модель.

5 Обеспечивающие системы (мониторинг, оркестрация, тесты)

1	**Оркестрация**	- Как организовать многоэтапные процессы обработки данных - Инструменты оркестрации - Oozie, Airflow
2	**Интеграция, тестирование, развертывание. CI / CD. DevOps.**	- Культура DevOps - Работа в команде - CI / CD - Auto tests
3	**Мониторинг**	- Инструменты мониторинга - Prometheus, Zabbix, Graphite, Grafana - Специфика мониторинга процессов обработки данных Домашние задания 1 Задание: развернуть и настроить инструменты мониторинга. Проанализировать текущие показатели.
4	**Data Quality. Контроль качества данных, мастер-данные, Troubleshooting.**	- Data Quality and Consistency. Качество данных. MDM - Ошибки в коде, логике, виды, последствия, как найти и устранить корневую причину - Вопросы поддержки. Support - Network, integration, data quality, system faults, disk space, executor out of memory, grants, access rights, security - Метрики качества. Контроль качества. Data Fix - как исправлять найденные ошибки - MDM: управление мастер-данными
5	**Case studies. Кейсы компаний.**	Углубленные вопросы оптимизации. Фишки. Примеры, разбор Домашние задания 1 Задание: разработать проверки качества данных для витрины. Внедрить их автоматическое выполнение. На предложенных примерах попытаться выполнить устранение ошибок (Data fix).
6	**Бонус. Дальнейшее развитие Hard skills + Soft skills.**	- Где искать ответы на вопросы. Ресурсы. Как быстро разбираться и решать проблемы. - Benchmarking - умеем сравнивать инструменты для решения конкретных задач - Как грамотно составить резюме (CV) + proof-read резюме участников курса - Как развиваться в плане Soft skills, Hard skills. Contribution. Домашние задания 1 Задание: подготовить резюме (CV), отрецензировать резюме товарища. Завести аккаунт в LinkedIn.

6 Проектная работа

1	**Вводное занятие по проектной работе**	Слушатели курса смогут определиться с темой проекта (выбрать из предложенного списка или привнести задачу из деятельности своей компании), получить понимание какие ресурсы им необходимо использовать для работы. Домашние задания 1 Проектная работа
2	**Консультация по проектной работе**	Слушатели курса получат комментарии относительно прогресса проектной работы, ответы на вопросы, рекомендации по реализации.
3	**Защита проектной работы**	По окончании занятия слушатели курса получат разбор проектов, комментарии и оценку своей работы.