**Data Engineering Essentials**

[API](https://skillbox.ru/media/code/chto_takoe_api/) – application programming interface. Список функций, позволяющих работать с данными, находящимися на сервере другой компании, устройства. Позволяет делиться данными, не раскрывая их внутреннее устройство и код.

[REST](https://habr.com/ru/articles/483202/) популярный архитектурный подход для создания API. HTTP является основным строительным блоком REST сервисов.

[HTTP](https://skillbox.ru/media/code/chto-takoe-http-i-zachem-on-nuzhen/) – протокол передачи гипертекста. Используется для взаимодействия клиента (часто браузера) с сервером.

[JSON](https://skillbox.ru/media/code/json-chto-eto-za-format-i-kak-s-nim-rabotat/) - стандартный текстовый формат для хранения структурированных данных и обмена ими. Стандарт индустрии для получения данных с веб-страниц. Заменил формат XML.

[Kafka](https://habr.com/ru/companies/sbermarket/articles/738634/) – позволяет сместиться с ООП на событийно-ориентированное программирование. Хранит данные в виде упорядоченном логе событий. Состоит из продюсера, брокера и консумера. Продюсер выдает данные, брокер обрабатывает, консумер использует. Позволяет достичь очень высокой быстроты выполнения операций. Имеет разные виды точности, может быть толерантен к дублирующимся данным или к их пропускам, в зависимости от задач

[Hadoop](https://practicum.yandex.ru/blog/gde-i-zachem-ispolzuetsya-hadoop/) – распараллеливание вычислений путем дробления задачи на подзадачи и их выполнение на несколько кластерах и узлах. Работает по технологии MapReduce.

[Data Warehouse](https://habr.com/ru/articles/441538/) – структурированная база данных, может иметь разные типы. Наиболее популярные это снежинка и звезда. В звезде центральная таблица (Fact Table) связана с остальными (таблицы измерений) только одной ссылкой. Например, центральная таблица содержит информацию о заказе (покупатель, продукт, и т.д.), тогда каждый столбец в этой таблице раскрывается в одну таблицу (например, покупатель является внешним ключом для таблицы покупатель, которая содержит информацию о его возрасте, городе и тд). В звезде, которая занимает меньше места, но требует написания более сложных запросов, связь осуществляется через систему таблиц. Например, столбец продукт таблицы фактов будет вести на таблицу бренд, которая в свою очередь будет вести на таблицу поставщик. Таким образом, уменьшается количество ненужной информации при запросе, но повышается сложность самого запроса.

1. Галактика (Galaxy Schema)

Архитектура галактики представляет собой комбинацию нескольких звездных схем. Эта архитектура используется, когда в системе данных необходимо поддерживать множество различных бизнес-процессов, каждый из которых имеет свои собственные факты, но с общими размерностями.

2. Трехслойная архитектура (Three-Tier Architecture)

Эта архитектура включает три уровня:

Источник данных (Data Sources): Сюда входят все системы, откуда извлекаются данные, включая базы данных, приложения и внешние данные.

Хранилище данных (Data Warehouse): Здесь данные централизуются и хранятся после ETL процессов.

Инструменты доступа и анализа данных (Data Access and Analysis Tools): Включает OLAP-сервисы, аналитические приложения и пользовательские интерфейсы для доступа к данным.

Подход Ральфа Кимбалла **основывается на важности витрин данных, которые являются хранилищами данных, принадлежащих конкретным направлениям бизнеса**. Проект хранилища данных по принципу Кимбалла использует подход «снизу вверх».

Подход Билла Инмона основывается на том, что хранилище данных является **централизованным хранилищем всех корпоративных данных**. Это известно как нисходящий подход к хранилищу данных.

[ETL](https://habr.com/ru/articles/695546/)– Extract, Transform, Load принцип. Подходит для хранения данных в реляционных (структурированных) базах данных, где для загрузки данных необходима их пред-обработка для приведения в вид, подходящий для существующей архитектуры.

[ClickHouse](https://habr.com/ru/companies/otus/articles/773174/) – столбчатая база данных, позволяющая быстро работать с большим объемом данных. Формат хранения данных в виде столбцов предполагает, что каждый столбец данных хранится независимо от других. Это позволяет системе выполнять сложные запросы в течение считанных секунд, запрашивая только несколько колонок. Система может обращаться только к тем столбцам, которые фактически требуются для выполнения запроса, и не обрабатывать данные в тех колонках, которые не участвуют в запросе

[Data Lake](https://practicum.yandex.ru/blog/chto-takoe-ozera-dannyh/) – озеро данных, в котором могут хранится данные разных форматах, метаданные этих данных, а также резервные копии. Нужно следить за состоянием, чтобы озеро не превратилось в болото.

[ELT](https://habr.com/ru/articles/695546/) – Extract, Load, Transform принцип. Подходит для хранения данных в Data Lake. Гораздо быстрее чем ETL, позволяет почти мгновенно загружать данные (обычно с использованием Hadoop). Манипуляции происходят непосредственно в озере данных. В дальнейшем, данные нужные для анализа могут быть извлечены в упорядоченную базу.

SQL Transaction – инициализируем pipeline к базе данных (connect), далее выполняем нужные запросы, если все как ожидалось, коммитим изменения (commit), если есть ошибки, откатываемся к исходным данным (rollback). Коннектор имеет пул подключений, поэтому важно закрывать запросы, чтобы сохранять ресурсы коннектора

[ACID](https://habr.com/ru/articles/555920/) – **atomicity** (операция либо выполняется полностью, либо, например, встретив ошибку, не выполняется вообще), **consistency** (база данных не должна содержать разночтений), **isolation** (параллельные транзакции не должны оказывать влияния на результат), **durability** (не может быть отката к прежнему состоянию, если пользователь оповещен о завершении операции)

[Airflow](https://practicum.yandex.ru/blog/apache-airflow/) - Архитектура Airflow базируется на концепции направленного ациклического графа (DAG). По факту составляет график задача и выполняет их согласно DAG. То есть не начинает следующую задачу, пока не разобрался с предыдущей.