**Data Workflow:**

**There are four general steps through which data flows within an organization:**

1. **Data collection & storage**: Collect and Ingest data, from web traffic, surveys, or media consumption for example.
2. **Preparation**: prepare it, which includes "cleaning data", for instance finding missing or duplicate values, and converting data into a more organized format.
3. **Exploration & Visualization**: explore, visualize, and build dashboards to track changes or compare two data sets.
4. **Experimentation & Prediction**: ready to run experiments, like evaluate which article title gets the most hits, or to build predictive models, for example, to forecast stock prices.

**Data Engineers:**

Data engineers are responsible for the first step of the process: ingesting collected data and storing it.

They are responsible for laying the groundwork for data analysts, data scientists and machine learning engineers.

If the data is scattered around, corrupted, and difficult to access, there's not much to prepare, explore, or experiment with.

And that's exactly why you need a Data engineer: their job is to **deliver the correct data, in the right form, to the right people, as efficiently as possible**.

They:

* Ingest data from different sources.
* Optimize the databases for analysis.
* Manage data corruption.

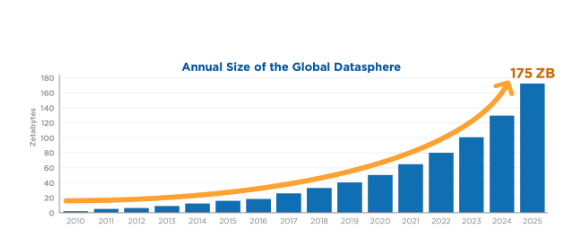
Data engineers develop, construct, test, and maintain architectures such as databases and large-scale processing systems to process and handle massive amounts of data.

**Big Data & Data Engineers:**

**With the advent of big data, the demand for data engineers has increased.**

Big data can be defined as data so large you have to think about how to deal with its size, because it's difficult to process using traditional data management methods.

This graph helps make sense of the growth of big data.

****

**The Five Vs:**

**Big data is commonly characterized by five Vs:**

1. **Volume**: the quantity of data points.
2. **Variety**: type and nature of the data (text, image, video, audio).
3. **Velocity**: how fast the data is generated and processed.
4. **Veracity**: how trustworthy the sources are.
5. **Value**: how actionable the data is.

Data engineers need to take all of this into consideration.

---------------------------------------------------------------------------------------------------------------------------------------------->>>

**Data engineers vs. data scientists:**

**Data Engineers** role is to ingest and store the data so it's easily accessible and ready to be analyzed.

**Data scientist** intervene on the rest of the workflow: they prepare the data according to their analysis needs, explore it, build insightful visualizations, and then run experiments or build predictive models.

Data engineers lay the groundwork that makes data science activity possible.

|  |  |
| --- | --- |
| Data Engineer | Data Scientist |
| * Ingest and store data * Data engineers ensure that databases are optimized for analysis (correct table structure, information easy to retrieve) * Build Data pipeline | * Exploit this stored data * Access the databases to exploit the data it contains, For example, can analyze data without too much preparation work. * Use pipeline outputs |

Based on the above, it's no surprise that data engineers are **software experts**, while data scientists are **analytics experts**. In general, **Data Engineers** uses languages like **software-oriented Python** or **Java**, and **SQL** to create, update and transform databases, while **Data scientist** uses **analytics-oriented Python** or **R**, and **SQL** to query - or, in other words, request information from - databases.

**Data Pipeline:**

Companies ingest data from many different sources, which needs to be processed and stored in various ways. To handle that, we need data pipelines that **efficiently automate the flow from one station to the next, so that data scientists can use up-to-date, accurate, relevant data.**

Data pipelines ensure the data flows efficiently through the organization.

**They** automate:

* extracting
* transforming
* combining
* validating
* loading data

To

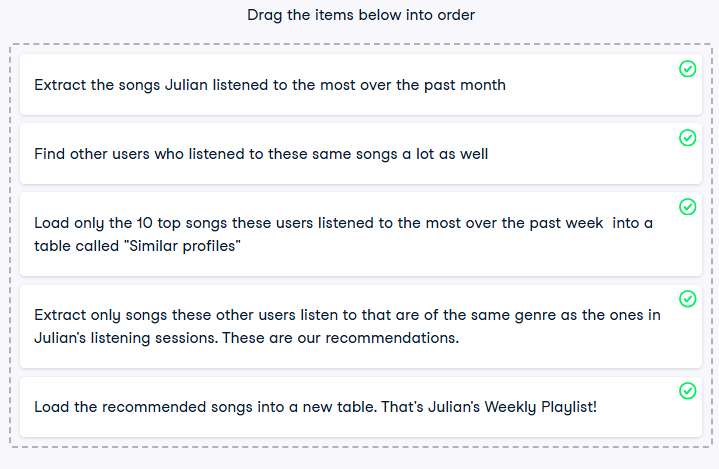
* Reduce human intervention and errors
* Decrease the time it takes for data to flow through the organization.

**ETL:**

It's a popular framework **for designing data pipelines**. It breaks up the flow of data into three sequential steps: first **E for extracting** the data, then **T for transforming** the data, and finally, **L for loading** this transformed data to a new database. The key here is that data is processed before it's stored.

In general, **data pipelines move data from one system to another**. They may follow ETL, but not all the time. For instance, the data may not be transformed, and routed directly to applications like visualization tools or Sales force.

Example:



---------------------------------------------------------------------------------------------------------------------------------------------->>>

**Data structure:**

* **Structured data**
  + Structured data is easy to search and organize.
  + Data is entered following a rigid structure, like a spreadsheet where there are set columns.
  + Each column takes values of a certain type, like text, data, or decimal.
  + It makes it easy to form relations; hence it's organized in what is called a relational database.
  + About 20% of the data is structured.
  + SQL, which stands for Structured Query Language, is used to query such data.
* **Semi-structured**
  + Semi-structured data resembles structured data, but allows more freedom.
  + It's therefore relatively easy to organize, and pretty structured, but allows more flexibility.
  + It also has different types and can be grouped to form relations, although this is not as straightforward as with structured data - you have to pay for that flexibility at some point.
  + Semi-structured data is stored in NoSQL databases (as opposed to SQL) and usually leverages the JSON, XML or YAML file formats.

**Example:**



Here is an example of a **JSON file** storing the favorite artists of each Spotflix user.

As you can see, the model is consistent: each user id contains the user's last and first name, and their favorite artists. **However, the number of favorite artists may differ**: I have four, Sara has two and Lis has three favorite artists. **Relational databases don't allow that kind of flexibility, but semi-structured formats let you do it.**

* **Unstructured data**
  + Unstructured data is data that does not follow a model and can't be contained in a rows and columns format.
  + This makes it difficult to search and organize.
  + It's usually text, sound, pictures or videos.
  + It's usually stored in data lakes, although it can also appear in data warehouses or databases.
  + Most of the data around us is unstructured.
  + Unstructured data can be extremely valuable, but because it's hard to search and organize, this value could not be extracted until recently, with the advent of machine learning and artificial intelligence.