**What is Data Engineering?**

[A large red bridge over a body of water

Description automatically generated with medium confidence](https://camo.githubusercontent.com/861ef81fbde2b60a0af37233be9c0147e0a6e65072b49dfadff22aed4c8a0d45/68747470733a2f2f736f757263652e756e73706c6173682e636f6d2f675a5878386c4b41623759)

**Data Engineering** is a relatively new field in data-related topics. When we are talking about data folks, we usually think about two types of responsibilities inside the team:

1. those who *analyze* the data
2. those who *prepare* the data for analysis

In the first category there are Data Analysts, Data Scientist, Data Visualization specialists, Product Analysts, etc.

In the second category there are more technical people: Data Engineers, BI developers, DevOps/DataOps.

In many companies this separation is not very strict, though. For example, some Data Scientist may do a job of a Data Engineers, some BI developers can step on a Data Analysts field, etc. But generally speaking, there are those who *work with the data* and those who *deliver the data*. There is a great quote from British engineer and author Gordon Lindsay Glegg:

*A scientist can discover a new star, but he cannot make one. He would have to ask an engineer to do it for him.*

And I cannot agree more.

**Why do we need Data Engineers, anyway?**

According to [some surveys](https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#51aacb106f63), Data Scientist can spend about **80% of time preparing the data**.

Now, if you take into consideration the cost of such specialists it becomes clear that it is a waste of time and money. Data Scientist and Data Analysts should spend more time analyzing the data, because it is their core value.

And here is where a Data Engineer is joining the game. Data Engineer is a tech-savvy specialist who can help data teams with many technical data-related topics.

**Areas of responsibilities**

Let's talk a bit about activities Data Engineers are responsible for. If we look at the famous [The Data Science Hierarchy of Needs](https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007), we can highlight the following areas:

1. **Data storage**. Data Engineers are responsible for building a reliable data storage to keep all the collected data. This topic includes maintaining databases, data warehouses, data lakes and other types of storages. There are different areas to take care of: where to store the data (in clouds or on-premise), which technologies and frameworks to choose (based on provided use-case), how to maintain and scale such systems.
2. **Data flows**. Data Engineers should know how to collect new data, what data is already available, and how to move the data between the systems. This is usually called as ETL (or ELT, the definition and the difference will be explained in later chapters). We need to move data from a source systems to a data warehouse, we need to build a data aggregates, we need to convert and prepare the data for different systems, and many other activities related to data moving and data transformation.
3. **Data monitoring**. Usually it is not enough to build a data pipeline, it is also important to make sure that it works seamlessly. All data pipelines should be monitored and alerted in case of issues. It will not only rise the quality of data, but also will increase confidence in such data.

**Bringing value**

One thing every Data Engineer should always remember – to **bring value** to the business. What is the value of a system which no one uses? What is the value of a system which is faulty and produces incorrect data? Right, there is no value for the business whatsoever.

But how you bring value as a Data Engineer? It's easy, just follow these three rules:

* **Deliver useful and reliable data services.** Everyone should to be confident in your work.
* **Know the client and the needs.** Talk to you client and find the value which he wants to get from you.
* **Deliver solutions which comply with the needs.** Deliver what people expect and even more.

**Cycle of delivering results**

These rules sound easy, but are not always easy in practice. To simplify your life, try to follow this **cycle** while you are working on a project:

1. **Clarify all requirements with the client.** Ask about the problem your client wants to solve and about the desired solution.
2. **Make a hypothesis or suggestion.** What client wants is not always what he *really wants*. Try to look at the problem broader, because as a technical person you may see things from a different perspective.
3. **Validate the hypothesis.** Try to build a first solution as simple as possible. Believe me, it will give you a lot of food for thoughts. Show this prototype to your client.
4. **Deliver the result, in iterations.** Given an experience gained from previous step, build an MVP. Gather the feedback and iterate a new (and better) version of your product.

**Communication**

And lastly, Data Engineers don't exist in vacuum, they need to communicate with other squads. In general, every DE should talk with:

* backend guys: software engineers, IT, data owners, security specialists
* own team: team lead, other data engineers, analysts, QA
* end-users and business people: data scientists and management

Don't forget to communicate. Tech people tend to avoid talking, but late communication may result in a wasted time and efforts.

**Summary**

I hope I didn't scare you with all that stuff and you want to continue your journey to Data Engineering world.

If so, head to the next chapter where I'm going to talk about [Data Engineering roadmap](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/1.2-data-engineering-roadmap.md) and describe skills of a good Data Engineer.

**Data Engineering roadmap**

In [previous chapter](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/1.1-what-is-data-engineering.md) we learned that data engineers work on delivering good data services. To build such services you will need to learn a lot because such job requires many cross-discipline knowledge.

To master Data Engineering (or DE for short) you need to learn many areas of Computer Science, from programming and hardware to the specific domain knowledge. But fear not, you don't have to learn all the concepts of the profession at once.

I like the lean approach to learn Data Engineering. In this roadmap I'll show you a way to level up gradually so you don't feel overwhelmed.

Let's talk about areas of expertise. A good Data Engineer should be proficient in the following:

[Diagram

Description automatically generated](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/img/fig-1-de-required-areas.png)

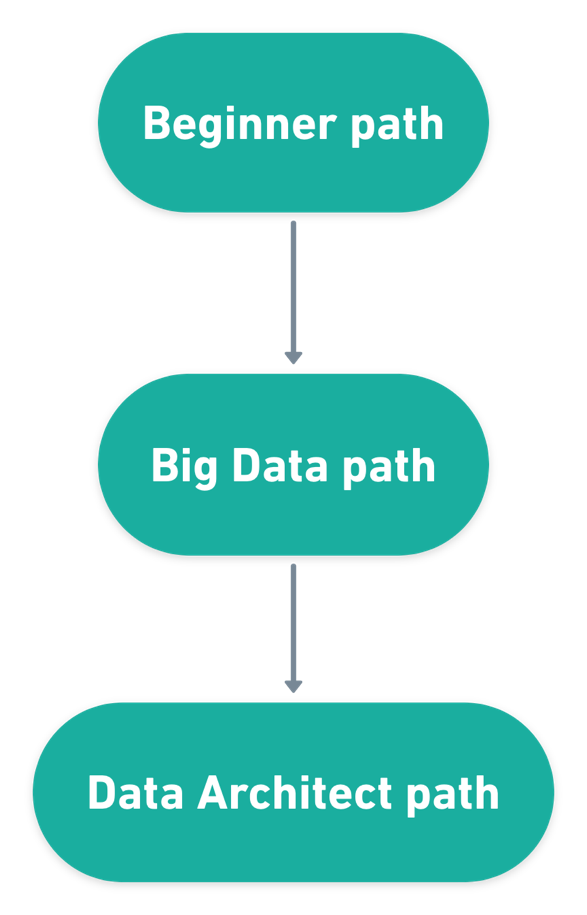
* **Software Development** – you need to know how to write good software
* **Tech Stack** – to understand and develop a good infrastructure
* **Analytics** – to speak the same language with analysts and the business
* **Domain Knowledge** – best practices of data engineering

So, my roadmap is based on those four areas. You need to gradually start learning all of them, advancing in each of them.

In the end I came up with the three paths of Data Engineering.

**Three paths of Data Engineering**

I like to split the roadmap into three path (or tiers): Beginner path, Big Data path and Data Architect path.

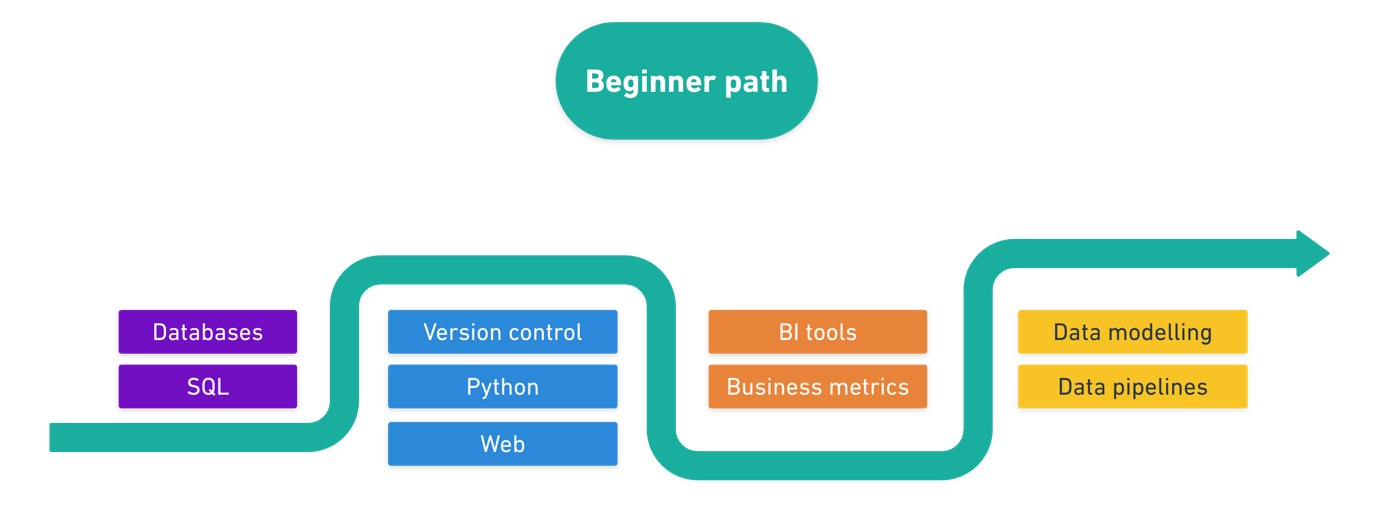
[](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/img/fig--high-level-roadmap.png)

This split is not a golden rule, but rather observation of the market. Those tiers are based on the job descriptions I've analyzed. For example, some DE positions require knowledge of databases, but not require knowledge of Big Data stack. Thus, such skills could be separated to different "buckets".

Another thing, you need think of those tiers as a layered pie: every new level should be on top the previous. Saying that, if you are an absolute zero in Data Engineering and start with Big Data path you will fail. As I said, higher levels are built on top of lower levels, so make sure that you understand concepts from lower tiers before starting the next one.

Let's look closely at each of the paths.

**Beginner path**

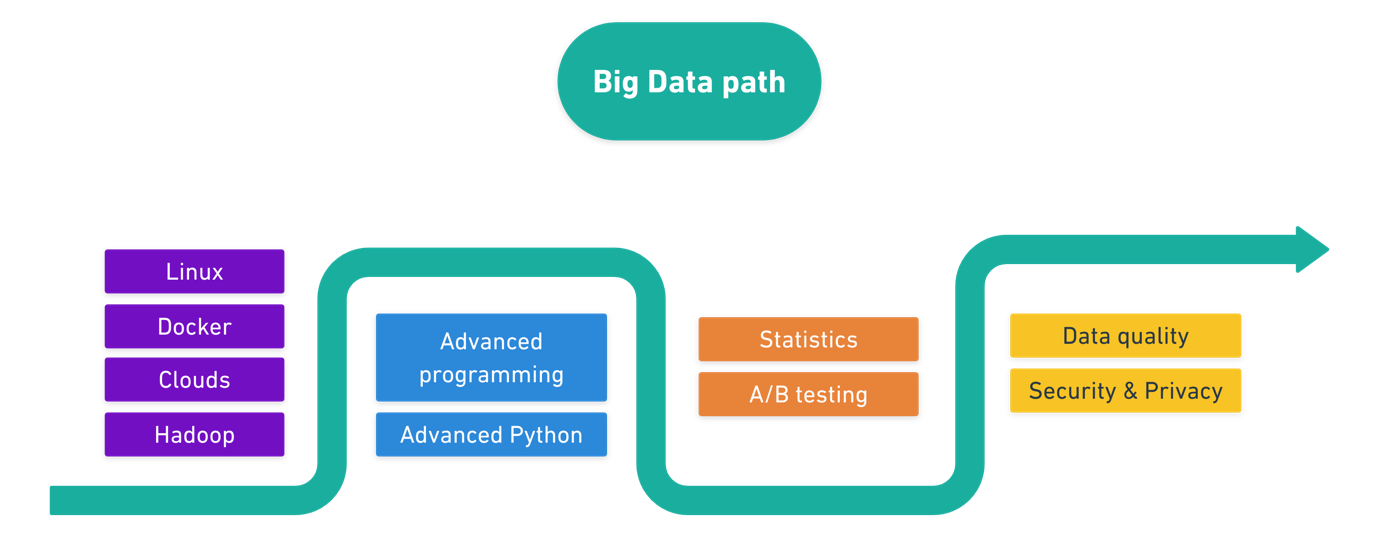
[](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/img/fig--beginner-path.png)

When you want to start DE career you need to work on some fundamental skill-sets.

Beginner path is all about data engineering basics. Here you need to gain a good understanding of SQL and relational databases, some programming experience (I'd recommend Python), basics of analytics and data engineering concepts.

Knowledge of the beginner level should guarantee you a job on Junior or Middle data engineering positions. After completing this level you will be able to communicate with databases, create data pipelines, model data for data warehouses and even visualize data in BI tools.

**Big Data path**

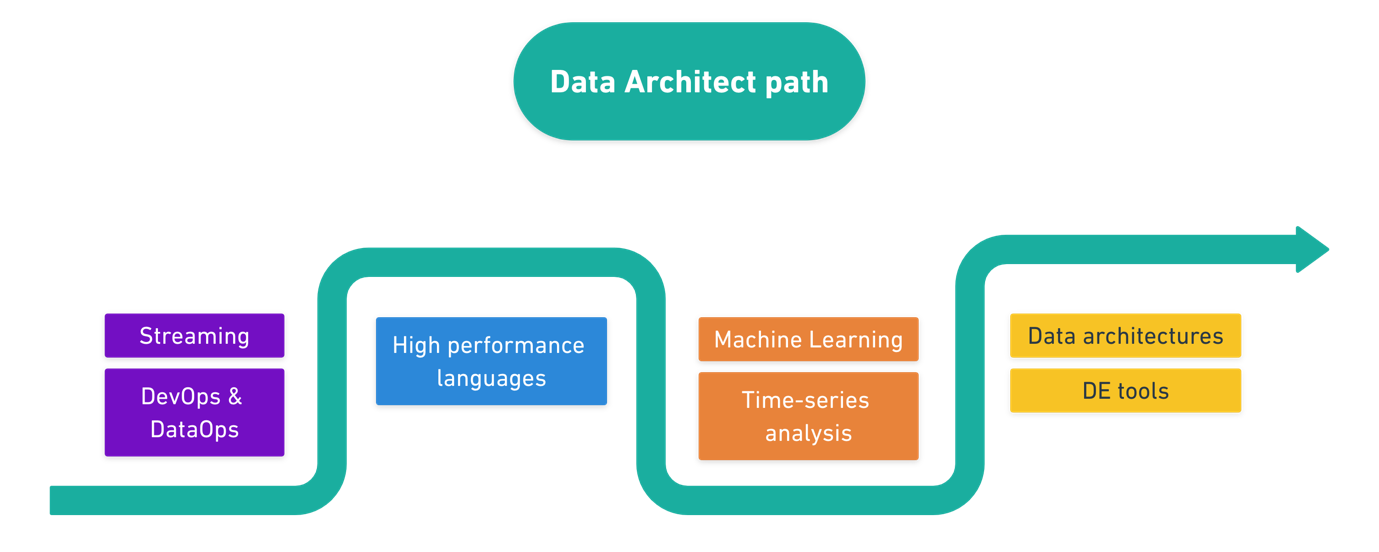
[](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/img/fig--big-data-path.png)

Big Data path is a continuation of the Beginner path with strong shift towards Big Data technologies. Probably it will take x2-x3 times more to master this path.

You should learn about Big Data stack, distributed processing, advanced programming, containerization, clouds, statistics, and continue with the most important data engineering concepts, like data quality and data security.

With knowledge of such level you are able to cover most of Senior Data Engineer requirements.

**Data Architect path**

[](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/img/fig--data-architect-path.png)

Data Architect path is a *High League*. People with such knowledge have a broad understanding of all Data Engineering concepts.

On this level you are not only able to create reliable data pipelines, but also decide the architecture decisions and pick the most suitable solution for each use case in data world. You will need to learn some advanced technologies like streaming and DevOps, learn more programming languages (like Java, Scala and Go), have hand-on experience with ML-engineering, and so on.

**Table of skills**

For convenience, I combined all skills and paths into one big table.

Come back to this tables once you feel that you need inspiration about "what to learn next?".

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Beginner path** | **Big Data path** | **Data Architect path** |
| **Software engineering** | * SQL * Version control * Python * Web | * Advanced programming * Advanced Python | * High performance languages |
| **Tech Stack** | * Databases * Relational and non-relational databases | * Linux * Docker * Clouds * Distributed processing * Hadoop stack | * Streaming * Operations (DevOps, DataOps) |
| **Analytics** | * BI tools * Business Metrics | * Statistics * A/B testing | * Machine learning * Time series analysis |
| **Domain knowledge** | * Data modeling * Data pipelines | * Data quality * Security and privacy | * Data architectures * DE tools landscape |

**Summary**

Data Engineering is a fast developing field and changes rapidly. New technologies emerge every day. That is why my roadmap focuses more on core skills rather than particular technologies.

Good luck with your journey!

**Table of content**

1. [Introduction to Data Engineering](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/README.md)
   1. [What is Data Engineering?](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/1.1-what-is-data-engineering.md)
   2. [Data Engineering Roadmap?](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/1-introduction-to-data-engineering/1.2-data-engineering-roadmap.md)
2. [Beginner path](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/2-beginner-path/README.md)
   1. [Intro to databases](https://github.com/oleg-agapov/data-engineering-book/blob/master/book/2-beginner-path/2-1-databases/databases.md)
3. Big Data path
4. Data Architect path