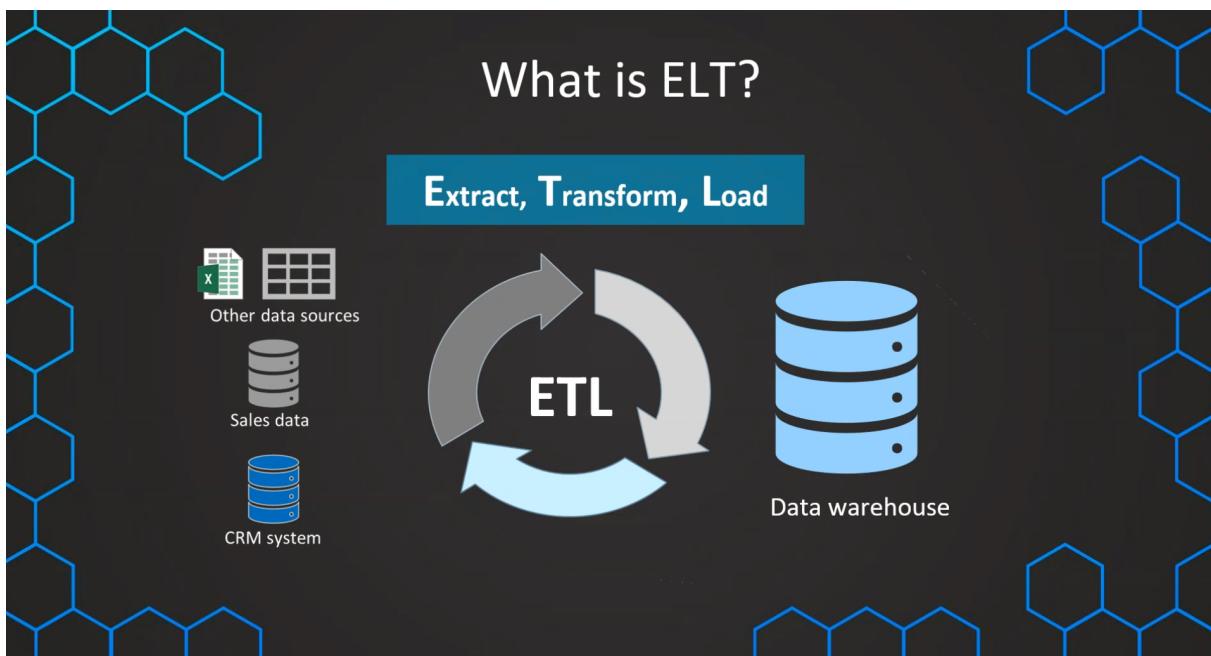
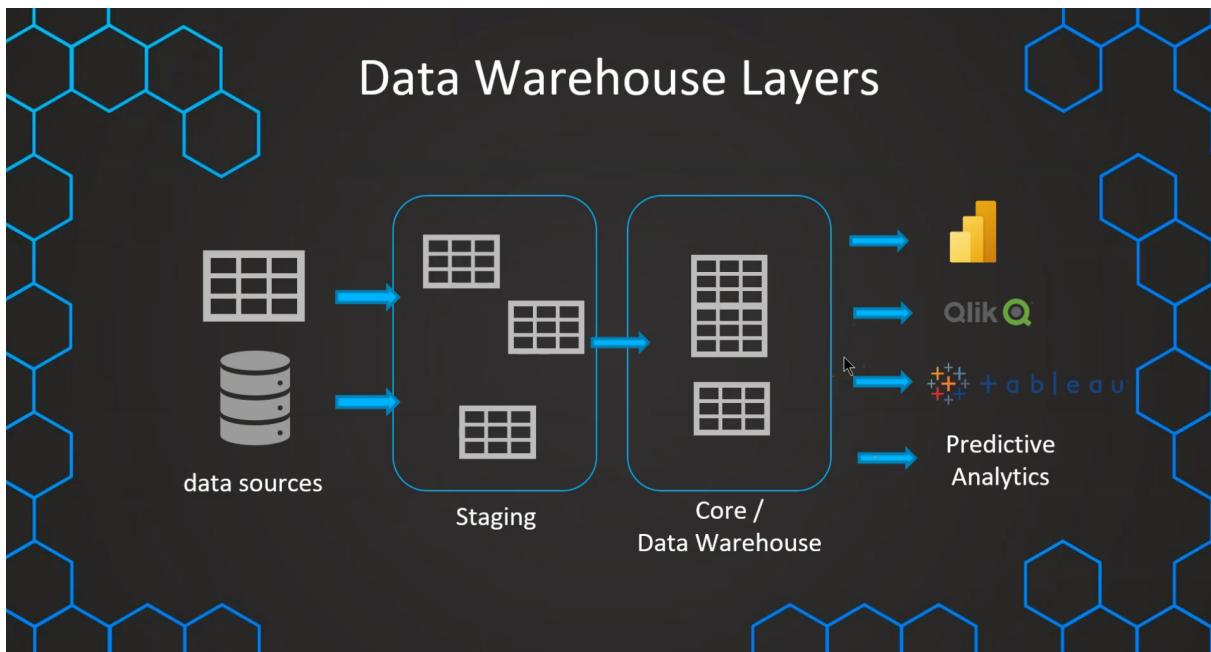


11. ETL vs ELT

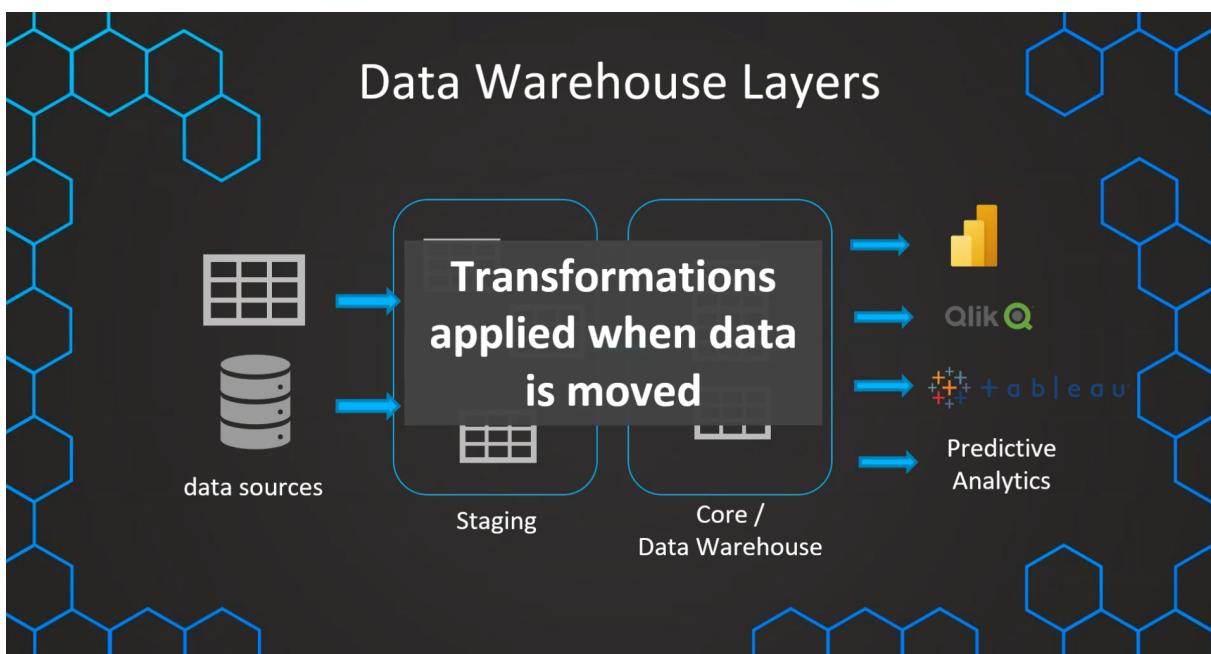
- There's also this term ELT, and this is a more modern approach, which is of course, not replacing an ETL, but it's worth looking at, because there can be some use cases, some scenarios and some advantages of this so-called ELT.



- So we've learned that the ETL is basically just the abbreviation of these three steps, which are extract, transform, and load our data.
- This means that in the beginning, we just extract the data to some staging layer, and then we do these transformations and with the data being transformed, we load it in our data warehouse and then of course, it is consumed by the different BI applications.

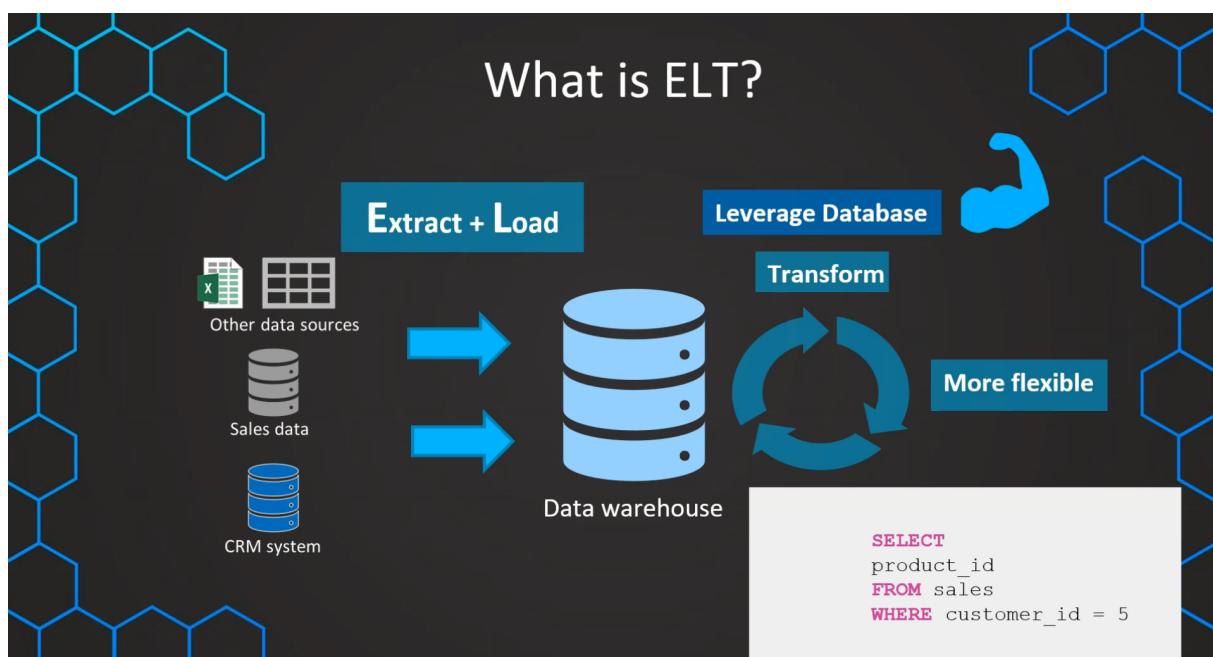


- This means that the transformations are basically applied when the data is moved and this is now different in the ELT.



- So in the ELT, we basically just switch around the order of loading and transforming.
- So in ELT, we first just extract the data, and load the data immediately in the data warehouse. And this is then, of course, not where we want to stop, because, we still want to apply our transformations.

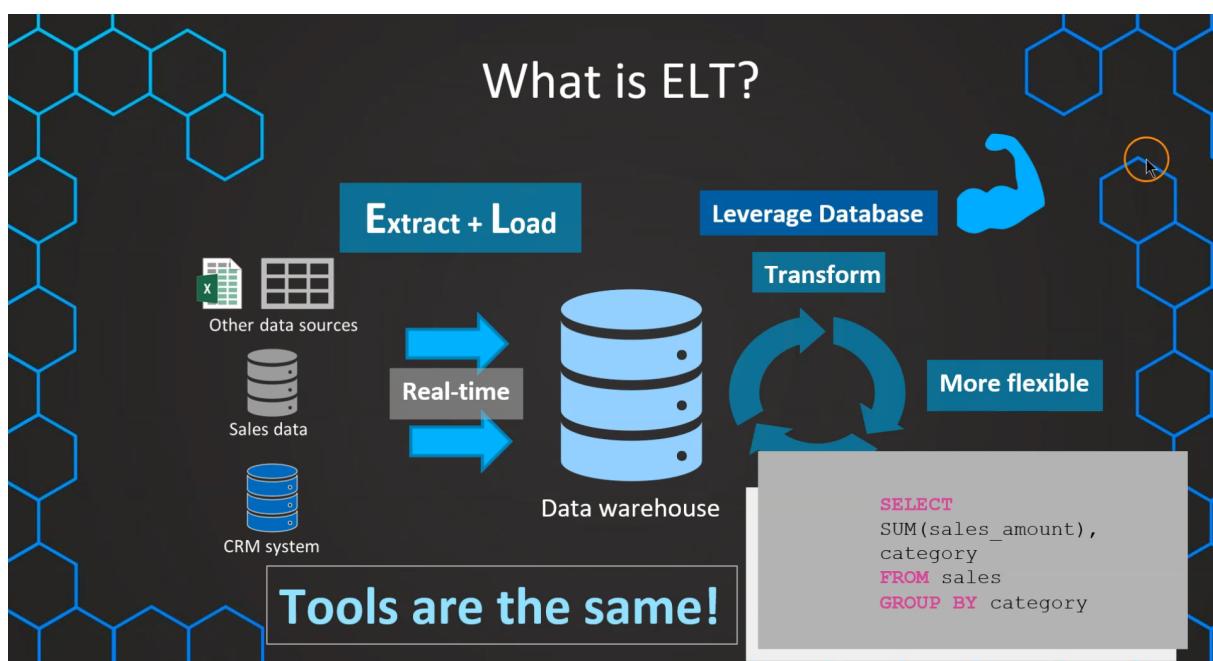
- And with the ELT, we do this using usually SQL commands in the database itself.
- This means that we need to leverage the compute power of the target database.
- So our data warehouse and this, of course, since we want to apply our transformations on the go, we require a lot more compute power.
- So with the rise of the cloud technologies, and cloud data warehouses, such as Snowflake, which can use massive parallel processing, which we'll touch on, of course, later on a bit too, we can now make use of that, and now apply our transformations just on the go, because we have all of that compute power.



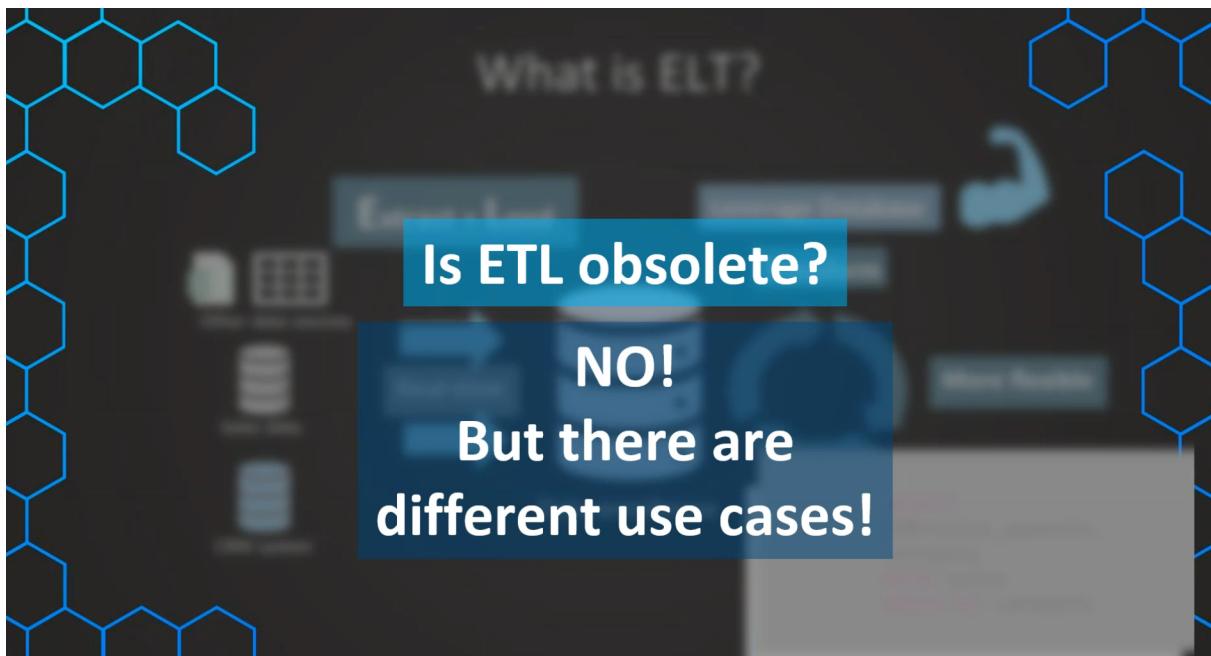
- And now, we are just more flexible with our transformations, because we don't need to plan them out in the beginning, but we can just apply them more flexibly on the go.
- And then every user, if we have multiple analysts, they can all run their own queries and use SQL for that. So they can, for example, also create views, and they can, just get the transformations more flexibly on the go.
- Another advantage is that now we don't need to transform our data before we load it, which makes the extraction process, and the loading much, much faster, because these transformations can be much more compute-intensive and this prevents us from loading our data, and running our ETL in

very high frequencies because this ETL can take more time, because we always need to apply those transformations.

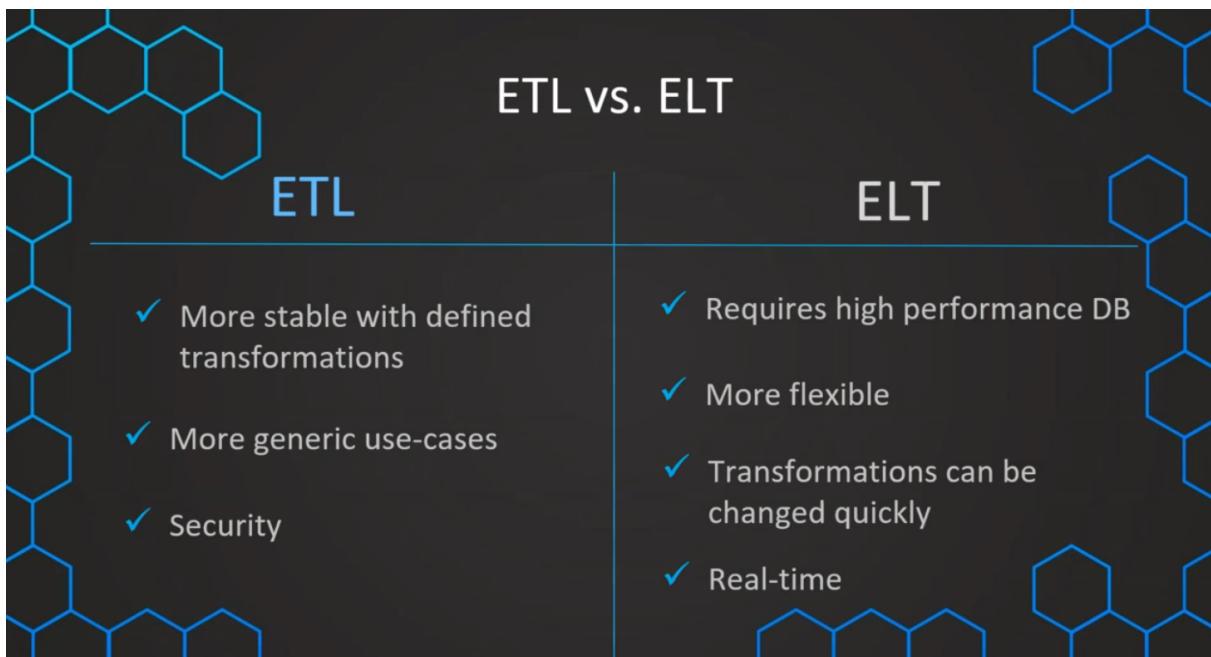
- But now if we just extract and load the data immediately, we can be really a lot, a lot faster.
- So if we need to have real-time data in our data warehouse, so whenever the data appears in the data sources, immediately, it should be loaded into the data warehouse. In such situations, we can also benefit from using an ELT instead of an ETL.
- So whenever we have some streaming data sources that immediately can get the data into our data warehouse, then this is a good choice to fulfill real-time requirements.
- Of course, we also want to note that the tool that we use can be really the same.



- So we can still use Pentaho, or other ETL tools that just extract, and load the data into our data warehouse. We just need to have a super powerful target database, and then we can benefit from this ELT.
- But now, the question, with these benefits, is an ETL now obsolete? And of course, the answer is NO. An ETL is still the most commonly used process to load, and transform the data into our data warehouse. But there can be some, in some situations, benefits and some use cases when we want to use an ELT instead of an ETL.

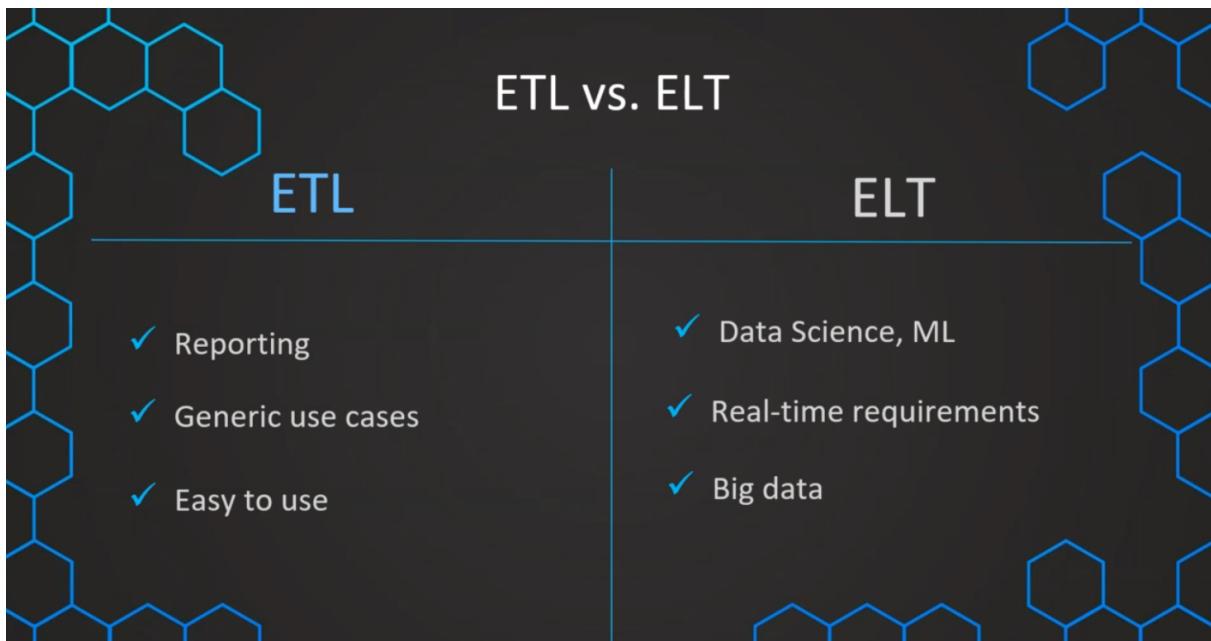


ETL vs ELT



- One very important requirement with that we cannot even use an ELT is that we need to have a very high performance for our target database.
- But now with these modern cloud data warehouses, we have that power sometimes. And in that cases, if we need to have this flexibility, we can consider an ELT instead of an ETL.

- With an ETL, we have just defined our transformations. The data will be loaded as we have just defined our transformations and with that, we have very stable data transformations.
- But in the ELT, which can be, of course, a benefit but when we are not very sure about what transformations we should use and we just want to be more flexible because the transformations could change from time to time, then we can also benefit from an ELT.
- So those transformations can be changed very quickly. We always have the raw data basically always available as it is and can just quickly on the go change our transformations.
- So this means also that in an ETL, we want to use it for generic use cases.
- So when we know our data should look like this, we don't have super specific and very different use cases, then we can just define our transformations and everyone can just use the data very easily as it is.
- Also, the security can be handled a lot better using ETL. So if we have some sensitive data in our ETL, we can, for example, only use the information that is not so sensitive so we can aggregate data and then in the data warehouse only that data lands there that is maybe already encrypted or anonymized. And with that, we can just more easily fulfill some security requirements.
- But with the ELT, we can fulfill more easily the real-time requirements because the process from extracting and loading is much less complex and it's just going a lot, a lot faster.



- So this is why we want to use the ETL usually for our reporting use cases.
- These are one of the main use cases we want to use a data warehouse anyways for and in such cases, we can just transform the data using ETL because we usually don't have real-time requirements and we have very defined transformation requirements.
- So this is why this is a good use case just for an ETL. But now sometimes,
- if we have some very sophisticated requirements from data scientists, they, for example, want to apply many different transformations that are very complex and they just want to dig a little bit in the data and they don't have very specific and generic transformations, then the company can also benefit from an ELT.
- And of course, also if we have real-time requirements, we should rather consider an ELT if the target database has enough performance, of course.
- A benefit of the ETL, on the other side, is that the resulting data in the data warehouse is just defined by the data engineers and then the resulting data is ready to use, which makes it very easy for the business users.
- In the ELT, we need to have more sophisticated data analysts that are transforming the data on the go.
- And this is, of course, also a difference. So also, if we have some big data, a lot of very high volume, then this can be sometimes a little bit too much to

handle for the ETL. In such cases, we can also consider an ELT. So these are the different use cases and the differences between ETL and ELT.