WHITE PAPER

Go Further with Snowflake's Data Cloud

INFOSTRUX

The purpose of this whitepaper is to provide a ground-up understanding of data analytics fundamentals.

We examine the different historical approaches that have been taken to implement data engineering. We look at traditional data warehousing and modern data warehousing, as well as the current trends such as Snowflake.

What is data analytics?

In short, data analytics is all about making data useful.

Fundementally, this means creating reports, dashboards, and KPIs to glean insights and make better business decisions – make the business more efficient and more productive.

On the more advanced side, it's about predictive analytics – using past data to forecast future trends, prescriptive analytics – making suggestions or taking actions based on past data and, and real-time analytics enabling businesses to respond quickly to changes.

What is data engineering?

While data analytics is the most visible discipline when it comes to realizing value from data, the value that can be extracted is intrinsically limited by the quality of the data analyzed. That is where data engineering comes in. Data engineering is about preparing data for analysis. This is done in five stages.

CENTRALIZING

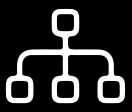
First, and perhaps most importantly, we must centralize the data by putting it in one place (whether physically or virtually) so that it can be made easily accessible for everyone. Data that is siloed or inaccessible by others doesn't unlock its full value.

CLEANING

Second, we have to clean it up. Quite often there are lots of data quality issues in the data – misspellings, null values, missing values, and many other data issues can cause incomplete or inaccurate data.

INTEGRATING

Third, we have to create consistent structures that align with rigorously defined business terms. This way everyone can rely on a single source of truth. It's very difficult to do data analytics on data that's not properly integrated.



MODELLING

Fourth, if the data is structured properly, business users can explore data, trends and create dashboards themselves without having to rely on analysts.

DOCUMENTING

Fifth, we have to document it. This way, anybody can go in and understand what they are looking at and where to find what they are looking for.

Why data engineering is so difficult

Data engineering is really quite difficult and requires lots of work.

Quite often, 80% of the effort is spent on data engineering simply because it is so time consuming and requires a lot of focus and attention.

The complexity of data engineering can be hard to perceive as it comes off as intuitively simple — "just" a matter of bringing data together and exposing it to end users. That couldn't be further from the truth.

To do it properly requires looking at many different aspects and fine details.

> Data engineering is time consuming, complex, and requires a lot of effort.

Understanding the business domain

Data engineers, or people who are trying to put together the data engineering solution, are most efficient when they can go in and understand every single detail about how the business works, how the business is structured.

For example, it would be quite difficult to put together a data engineering solution for a point-of-sale system if one doesn't understand the difference between the returns, refunds, and voids, when they can happen and when they cannot, how they impact the actual cash in the till, the amount of products sold, etc.

This is just the first part, but it's nowhere near the most difficult.

Understanding the requirements

As with any software project, understanding the requirements is paramount to getting it done right. For this, we have to go deep into the details.

Even a simple statement like ‘revenue by month’ can be interpreted in many different ways. Bringing up 'revenue by month' with a particular business user will mean something very clear to them. That meaning, however, might be
completely different if you ask someone from a different business function.

Revenue by month for Sales may mean booked revenue; for Finance it may mean collected revenue; for distribution it may mean invoiced revenue based on how much product has shipped.

That's just 'revenue', but you can see there are a number of variations.

The same issue arises with 'month'. What can be ambiguous about month? Again, depending on who we are talking to, this can be month based on the posted date, month based on the invoice date, month based on the book date, month from a fiscal calendar, month from the regular calendar, month from a manufacturing calendar, and so on.

Understanding the data

The next part is what requires the most time and effort. Quite often, a data engineer will be faced with data sources with hundreds or thousands of tables, each with tens or sometimes hundreds of columns. We have to understand perhaps not all of it, but a good portion of it. What does status 'activated' mean? Why is '9999-12-31' showing up in the data column?

There are lots of data points that have to be analyzed and properly comprehended before we can start working with the data.

Understand the Business Domain

Returns vs Refunds vs Voids

Understand the Requirements

Revenue by month

Understand the Data

What does 'status activated' mean?
What does '9999-12-31' mean?

Development

Then we have to go and do the work and develop these pipelines to make sure that the data gets moving. We will also need some software development expertise to pull the data from exotic sources or strange formats, from unusual APIs and so on.

Data architecture and modelling

The expertise to do data engineering properly requires knowledge in data architecture and data modeling. We have to know how to design the data pipelines properly so that they will work and scale properly with large volumes of data.

Business analysis

Producing proper requirements requires business analysis skills. A good business analyst already knows many of the intricacies of a business domain and asks the right questions to quickly understand how your business works.

Project management

Putting it all together becomes a project management task. It requires expertise from many different areas and then ensuring everybody talks to each other, all the data is captured, and everything is done properly.

Infrastructure and operations

On the infrastructure side, on the lower level of the whole solution we have to have very capable database administrators that are able to optimize the solution properly. We need to understand the infrastructure and operations, or at least have a DevOps process in place for managing updates and changes, which is quite often even more demanding than the infrastructural operations may be.

Data governance

Data governance is one of the very important parts of data engineering. We have to understand the policies under which data can be used. This is about who can see them, what the data is, where does it live, how you can move through the enterprise, what happens if there are sensitive parts of data, what exactly are the compliance requirements, and so on. Basically, anything policy related goes under data governance.

Data security

These days, we see the news about companies being hacked and their data being exposed on the internet. Of course, no company wants this to happen, therefore to make the solution secure, proper expertise is required to implement the right controls to protect the data.

3 main approaches to data engineering

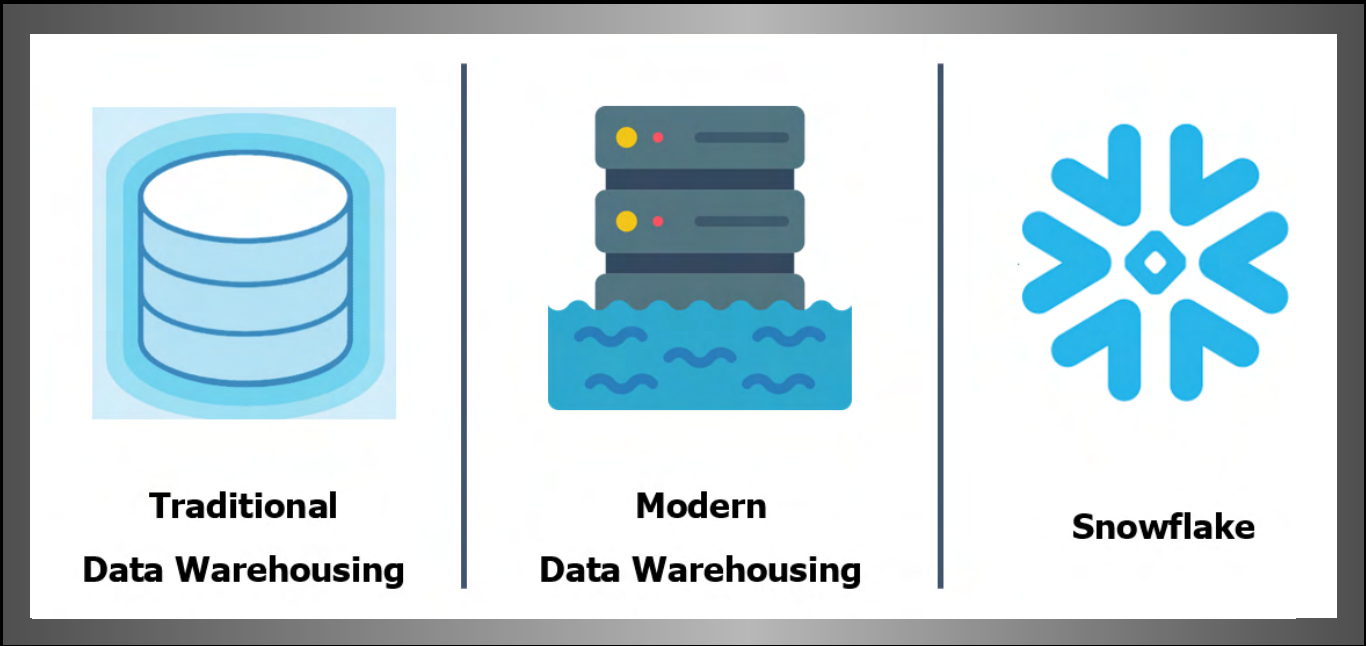
There are three main approaches we see in data engineering.

Prior to 2010, the majority of the data analytics solutions were done using traditional data warehousing.

That evolved into modern data warehousing, which does things a little differently and expands the capabilities of traditional data warehousing.

The current approaches move towards consolidation of different components and technologies of modern data warehousing into unified cloud-based platforms. An example of this approach is Snowflake.

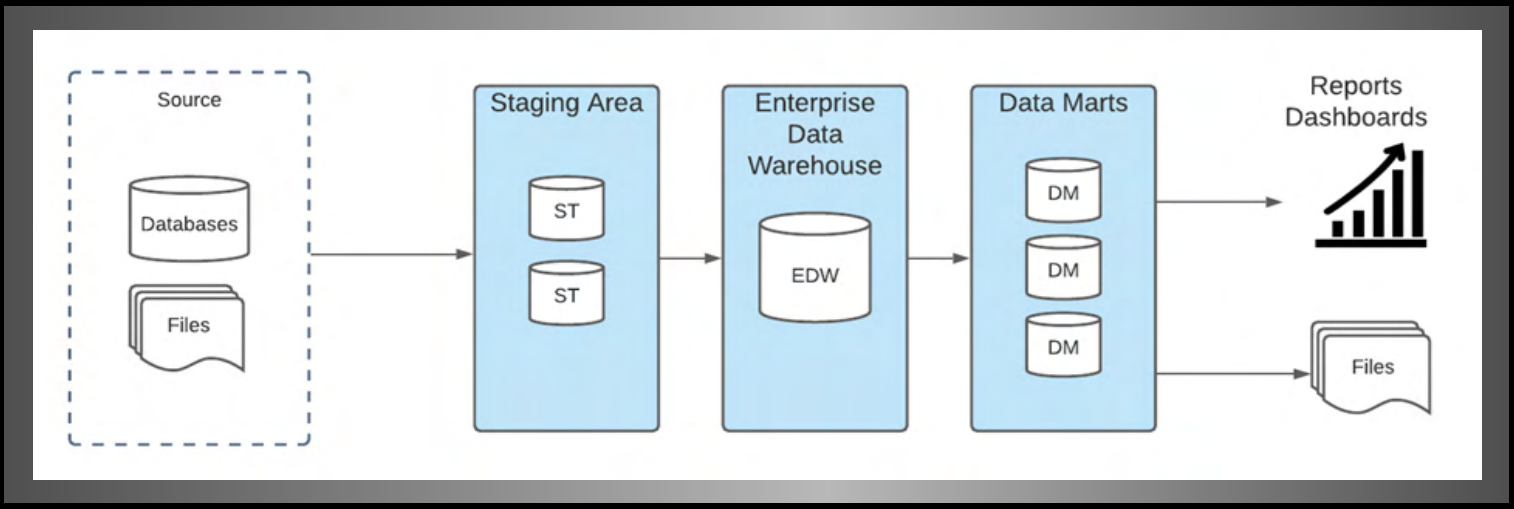
In this next section, we'll take a look at all three approaches.



Traditional data warehousing

Data sources addressed by traditional data warehousing are usually relational databases or structured files. From there, the data is loaded into a staging area, which contains tables that are structured similarly to the structure of the sources. It definitely does not contain all the data, quite often it contains only the differences between the last load and the current state of data in the data sources.

Once the data is loaded into the staging area, it's modelled into an enterprise data warehouse. This is usually a fairly normalized structure that reflects how the business works. Then the data is transformed into data marts that prepare the data for analysis through reports and dashboards, or exports.



The traditional data warehousing is done on a single system, using one or two major technologies. This could be SQL Server and SSIS, Oracle and Informatica or a similar pair. People working on the project would need to know only two technologies.

Challenges with traditional data warehousing

There are a few issues with the traditional data warehousing, the main issue is the data pipeline design and development.

Effort

In order to make a piece of data available for further analytics, it needs to be analyzed and understood, it has to be extracted to staging, it has to be cleaned, it has to be normalized, it has to be loaded into a data warehouse, dimensionally modeled, loaded into the data mart. If the data warehouse has hundreds of columns and thousands of rows, we need to do all of this a thousand times to ensure the solution works properly.

Expertise

That's where most of the effort is coming in. In terms of breadth of expertise needed to work on the project, we have to understand perhaps one single domain or one or two major data architecture or data modelling approaches. Even in the area where the effort is very high, which is the data pipeline design and development, we need to understand a couple of technologies.

	Effort	**Expertise**
Business Domain Understanding	low	narrow
Detailed Requirements Analysis	low	narrow
Data Architecture and Data Modelling	medium	narrow
Data Pipeline Design and Development	very high	narrow
Software Development	low	narrow
Database Administration and Optimization	medium	narrow
Infrastructure / DevOps	medium	narrow
Security	low	narrow
Data Governance	low	narrow
Project Management	low	narrow

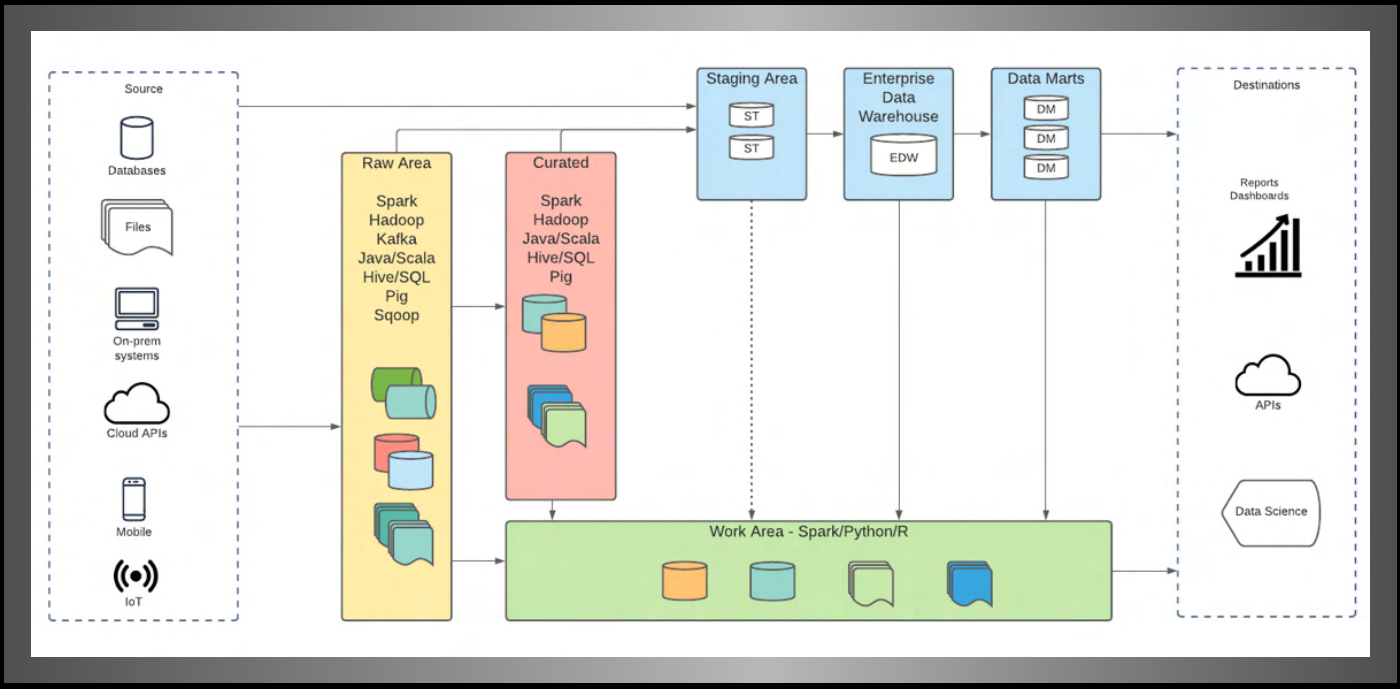
Traditional Warehousing

Each individual piece of data has to be:

- Analyzed and understood
- Extracted to staging
- Cleaned
- Normalized
- Loaded into a Enterprise Data Warehouse
- Dimensionally modelled
- Loaded into Data Mart

Modern data warehousing

There are different ways to describe modern data warehousing. The following diagram is just one example of how it can work.



With modern data warehousing, the sources are much broader. We still have databases and files as we had before, but we also have cloud APIs, mobile data, IoT data, on-premise systems, etc.

As a result of big data technologies that are now being used by modern data warehousing, we can afford to load all the data that we can think of, and store it for future use.

This was not possible with the traditional data warehousing where the staging areas were very limited and only a portion of the data can be stored.

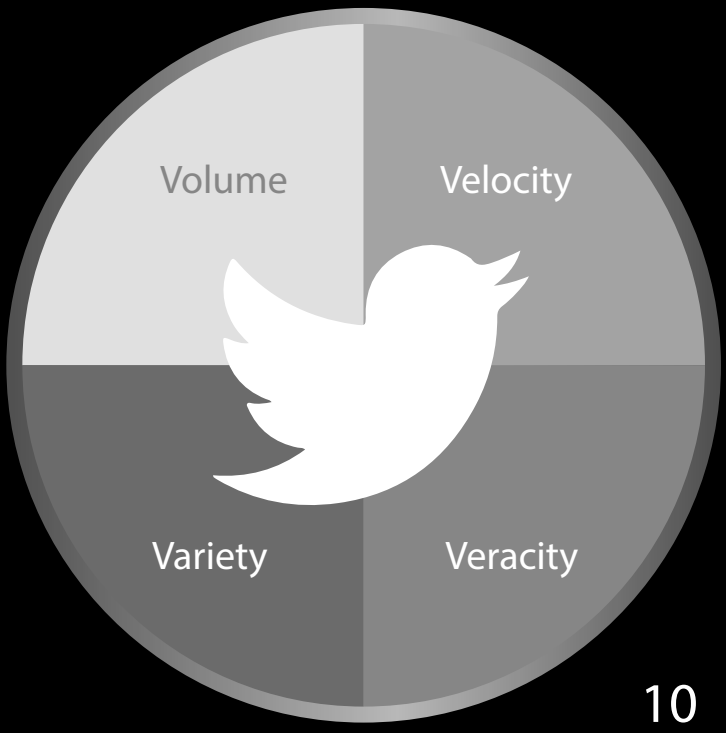
So modern data warehousing brings a good amount of new functionality to the table. There are a couple of things that make this possible.

> As a result of big data technologies, we can handle high volumes high variety, and high velocity of data.

Suddenly, we can do ELT as opposed to ETL. In other words, we can load the data before we transform it or touch it. The piece of technology that enables this is schema-on-load.

For example, Twitter data is a JSON structure that when it gets normalized ends up creating close to a hundred tables. That doesn't have to happen with modern data warehousing. We can simply keep the data as JSON, load it, and then look at the bits and pieces later if they become important or desired, but we don't actually have to come up and start staging them the same way as we would do in the traditional data warehousing.

So we're already saving ourselves the work of creating a hundred tables to make sure if we ever want to have access to all the Twitter data.



This enables data analyst's and data scientists' workflows. Suddenly, the data analysts and data scientists don't have to wait for your BI department or for the data engineers to bring data in all the way through the staging, all the way through the data warehouse, all the way to data marts, etc. They can directly hit the raw area that contains the Tweets and work with them in JSON format.

The effort required to enable a data scientist to come in and do their analysis becomes much smaller. We already have the data, the tools are there, they can handle the unstructured data just fine, the data scientists can come in and start working.

This also impacts the enterprise data warehouse and data mart areas of the solution. They can be smaller. We don't have to address every single request we get from data analysts and data scientists. We can keep the data in the data warehouse and data mart more narrow.

The data warehouses and data marts are still important. They still have to be there to provide the single source of truth that's correctly modelled and understood, ensuring 'revenue by month' means exactly what it needs to mean and the reports and dashboards are running properly.

When we look at the benefits of modern data warehousing, it can do more and hold different kinds of data. Another benefit is that there is no need to model everything. The effort that goes into building the model structure is lower.

But it has some drawbacks.

Challenges with modern data warehousing

Modern data warehousing is not without its challenges. This is no longer a single platform. As you saw in the above figure, there are many technologies and areas, making the system more complex to manage.

Another challenge is that it needs stronger data governance than before.

The amount of data that's available to the users is staggering. Therefore, strict policies are needed to maintain and govern it.

Without strong data governance, your data lake becomes a data swamp that nobody understands.

Generally, this would not be a problem with traditional data warehousing where everything is modelled and everything is prepared for analysis, but in modern data warehousing, this is definitely something that needs to be looked at.

	Effort	**Expertise**
Business Domain Understanding	low	narrow
Detailed Requirements Analysis	low	narrow
Data Architecture and Data Modelling	medium/low	wide
Data Pipeline Design and Development	high	wide
Software Development	low	narrow
Database Administration and Optimization	medium	wide
Infrastructure / DevOps	medium	wide
Security	medium	wide
Data Governance	medium	wide
Project Management	medium	narrow

In the table above, notice the data pipeline design and development went from very high to high. This is one of the important contributions of modern data warehousing. Not everything needs to be modelled and put into structured formats, and the fact that we can still deal with high volumes of the data of a high variety is really the most important benefit.

The data architecture and data modelling go down from medium to medium/low.

While these parts go down, many of the areas of expertise become wider.

Security suddenly becomes more of a concern because we are not securing a single database or a single database system that has one or two technologies in it. We are securing a hybrid of complex systems of many different parts, many different technologies, each with their own security requirements, needing their own security approaches, and they all have to work together in order to make the solutions secure.

Communications issues are becoming more prevalent. What quite often happens is that the traditional people who work on SQL don't talk to new people that are taking care of big data frameworks, and vice versa. The expertise needs to be added, teams now need to become broader.

Can we fix it?

To some degree, yes.

The tools we are working on these days offer more capabilities and they quite often run on the cloud. This leads to simplification of the whole data engineering stack.

The Snowflake data cloud is just one of the many cloud technologies that are providing more capabilities and simplification.

more capabiliies
+
cloud
=
simplification

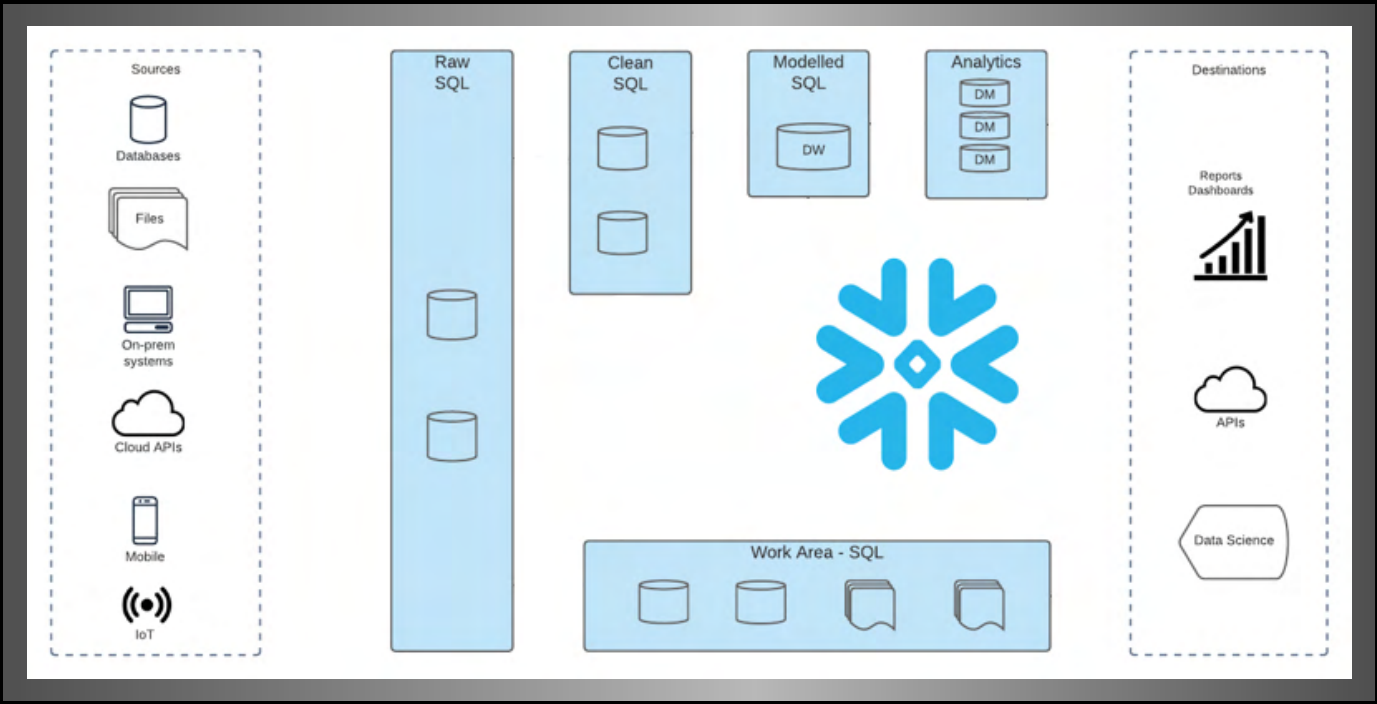
Snowflake Data Cloud

The Snowflake solution looks very similar to our modern data warehousing solution where we have the raw area and we can afford to load all the data with high velocity, high variety, and high volumes without much trouble.

The same platform can also implement a clean area where the raw data is transformed to more curated datasets.

It has a modelled SQL area where we can do data warehousing and whatever particular methodology we choose to go for. We have the analytics area, which can be materialized or virtual, depending on the approach. The work still needs to be done in order to set up the structures.

We also have the work area for our data scientists, which can hit any part of the solution.



In terms of functionality, it is very similar to modern data warehousing, but what's different now is that it's all done with one single tool. The expertise required goes down dramatically, and it's all SQL, which is a common language for anybody who works on the data engineering side.

The other benefit is that the solution comes fully managed. This is the cloud part.

We don't have to take care of the infostructure. Even the database administration and optimization is all done by the service that's capable of handling that part of the solution.

The other benefit that we got from modern data warehousing is that we no longer need to model everything anymore. It's still there and it's in the exact same way as it was before. We can just load all the data without a regard on how the schemas look. We simply store them in one simple solution and then grant access to the data analysts or data scientists, and anybody else who may need it.

Benefits

- No need to model everything
- One expertise - SQL
- Fully managed

The Snowflake scorecard looks better compared to traditional warehousing. Suddenly, the expertise required comes down again. It's one system, it's all SQL, it's fairly simple to do. In terms of effort, we are still gaining the benefits of modern data warehousing in that we don't have to model everything.

Currently, the data pipeline design and development is still high, but it's not very high as it was traditional data warehousing.

The data architecture and data modelling are medium/low as opposed to just medium.

	Effort	**Expertise**
Business Domain Understanding	low	narrow
Detailed Requirements Analysis	low	narrow
Data Architecture and Data Modelling	medium/low	narrow
Data Pipeline Design and Development	high	narrow
Software Development	low	narrow
Database Administration and Optimization	low	narrow
Infrastructure / DevOps	low	narrow
Security	low	narrow
Data Governance	medium	narrow
Project Management	low	narrow

The infrastructure / DevOps is getting lower on the effort side. This is because the service is running in the cloud and the infrastructure is actually hidden from the service users.

Database administration and optimization effort is getting low. Other platforms such as DataBricks, AWS Redshift or Azure Synapse, are all moving in the same direction, trying to expand their capabilities so that they can handle much more variety of workflows on the same platform without much infostructure or administration.

Data governance requires medium effort. We can still end up with data swamps if we are not careful.

So overall, it is getting better. It's not a silver bullet, the effort is still there, especially for traditional data analytics solutions where the main purpose of the exercise is to come up with your revenue by month reporting, and orders by product, and delivery by date.

Those still have to be done, but again the data analytics and the data science workflows are more enabled without doing much more effort.

Is it still difficult? Absolutely, it still requires a lot of work. We still have to go into the details, we still have to have a good amount of expertise, we still have to understand the domain, understand the requirements and understand data, but it is getting easier thanks to the technologies that take care of the concerns like infrastructure and the administration, as well as the technology that no longer requires us to model every single piece of data.

The overall effort of data engineering is moving in the right direction.

Thank you for reading our white paper. For more information about how Infostrux can provide data engineering solutions for you, please reach out to us today at info@infostrux.com, or visit www.infostrux.com.