Data Modelling in Big Data Solutions – why bother?

When I was a teenager, I read a quote that stayed with me. It was a comment by a character in a novel by David Eddings:

> "All the books in the world won't help you if they're just piled up in a heap."

It made me think about the importance of librarians and cataloging, in a way I never had before.

These days I work with Big Data. The tools we have for searching through text are far more powerful than in the pre-digital age. But the flip side is, now we have so much MORE data.

When very cheap storage became available, the concept of a "Data Lake" was formed, where all your data could be dropped and kept, for use later. Pretty quickly many organisations realised that what they had was not so much a "Data Lake", full of useful information, but a "Data Swamp" where information disappeared into the murk and was never seen again.

What is Data Modelling?

Modelling is the process of designing how data will be arranged and stored in your system, so end users can make use of it. If you don't invest in Modelling up front, at the beginning of your project, you will quickly find you have a Data Swamp, full of data you can't find or use easily.

Your data model design must answer many questions and balance competing needs.

For example:

What do you want the data store to do?

- Is this an analytical system, where data needs to be aggregated in diverse ways to produce reports?
- Is the data needed for use by Machine Learning?
- Does the data contain sensitive information that needs to be kept private, such as ID documents, or company financial data?

In many cases it's all of these and more, and distinct parts of your architecture will need to handle the data in different ways.

Some other important questions are:

- How much data do you have and how fast is your data volume growing?
- How quickly does data need to be processed? Do you need near instantaneous updates, or will overnight processing be OK?
- Are the people accessing the data experienced data analysts, or business users whose expertise lies in other areas?

This leads to decisions not just about the data model, but also about which toolset you need.

Different software products have different strengths and weaknesses.

Traditional database technologies such as SQL Server are first rate for smaller data sizes but struggle with the data volumes that large modern companies generate.

Big data technologies such as Spark and Hadoop can handle extremely large volumes of data but may be inefficient and poor value for money with smaller data volumes.

Data modelling levels

There are three levels of detail that you can use for data modelling.

Conceptual Model - This is a very high-level view of your data, broken into subject or domain areas.
Logical Model -This provides more detail and will describe individual tables and the relationships, business rules and constraints between them.
Physical Model – The most detailed version of the data model, with details of fields in tables, including data types and constraints. A Physical Model can be very valuable in a Spark implementation, to document the relationships between tables, even though those relationships are not enforced.

Don't let perfect be the enemy of good!

Often your model needs to be somewhat generic, to allow for changes in toolset later, but if you go too far with being generic, and trying to future proof, you won't get good performance from any software.

You will never be able to produce a 100% final design, because modern businesses change too quickly. A good initial design will set you up for success, so long as you remember to iterate and adjust to meet changing needs.

Data Modelling standards

The question of how to model and structure data in a data warehouse is a longstanding one. There are several well-established data modelling standards. These include:

- Inmon - Normalised
- Kimball – Star Schemas
- Data Vault – Hubs/Links/Satellites

Each standard has strengths and weaknesses, and determining the best fit for your project requires expertise and experience.

Can you use traditional data modelling standards in a Big Data solution, such as Spark?

Yes. But you must consider some differences in approach. For example: Spark does not enforce referential integrity between tables. This needs to be explicitly managed by your data ingestion process.

Conclusion

A well-designed data model in a big data solution will:

- Improve performance,
- Make the solution easier to extend,
- Make the solution easier to support,
- Enable easier use of data by end users – e.g., self-service reporting, data science.

Fujitsu Data & AI specialises in Data and can help you with designing and implementing Data Models to suit your project. Please contact a Fujitsu Data & AI specialist now.

Contact
Fujitsu Data & AI
+61 3 9924 3000