

10 Pitfalls to Avoid with Data Warehouses

Shôn Ellerton, October 2, 2019

Don't get caught out on some of the pitfalls of data warehouse design. Avoiding these 10 common pitfalls can save one a lot of time and money.

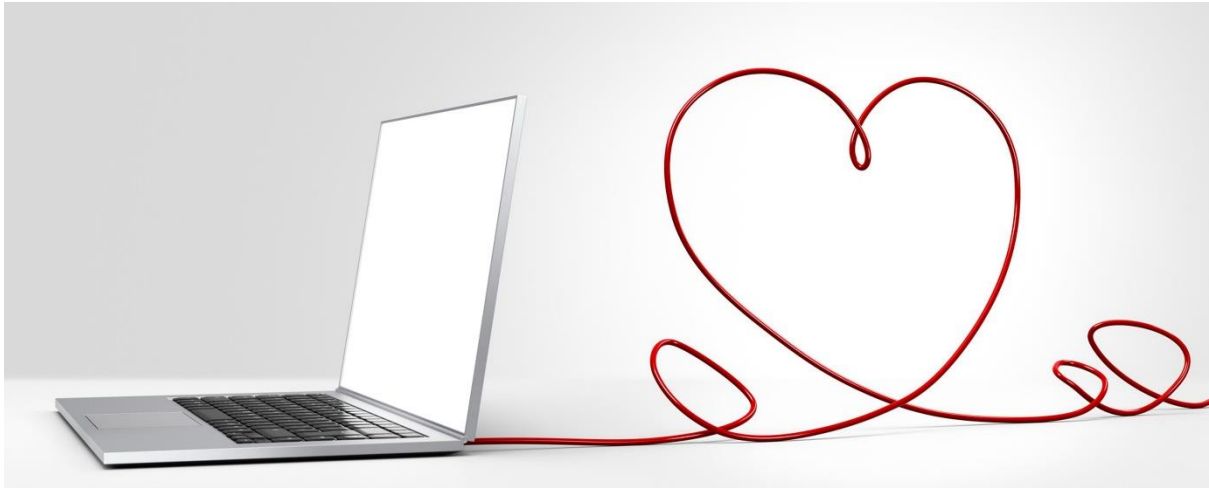


While thumbing through the book, *Data Warehouse Toolkit* by Ralph Kimball and Margy Ross, I came across 10 common pitfalls in data warehouse design which the authors collected together from other related articles. Against each one of the pitfalls, I added an additional comment related to the experiences I've had with data warehouse design.

Before embarking on a data warehouse project, I recommend having a quick read of these 10 common pitfalls of data warehouse design. It could save a lot of time and money.

Pitfall 10

'Become overly enamoured with technology and data rather than focussing on the business's requirements and goals.'



I've recently had two back-to-back contract jobs working as a data specialist. On one of the jobs, I worked on a project seemingly not going very far at all simply because those in charge of the project kept chopping and changing data architectures, methodologies and technologies to attain perfection without really understanding what the client wants. On the other job, we were simply allowed 'to get on with it' by understanding the needs of the business customers and implementing the solutions that they need to get on with their jobs. It wasn't perfect by any means, but who cares if it works.

Pitfall 9

'Fail to embrace or recruit an influential, accessible, and reasonable management visionary as the business sponsor of the data warehouse.'



Without a prominent business sponsor, most database solutions are likely to fail just as with any product. Lack of vision, enthusiasm, support, leadership and the ability to influence the business to invest in any product whether it is a new data warehouse or a robotic bath rubber duck is usually fatal.

Pitfall 8

‘Tackle a galactic multiyear project rather than pursuing more manageable, while still compelling, iterative development efforts.’



In my experience, this is the one pitfall which I see time and time again. Huge monolithic projects with battalions of siloed IT specialists working within constrained methodologies to create that one all-encompassing system that can not only tackle every report known to mankind but be able to make a cup of tea as well. To add insult to injury, those managing these projects tend not to consult with the underlying business in near enough detail to realise a useful product. Many a white elephant has been born in this fashion.

Pitfall 7

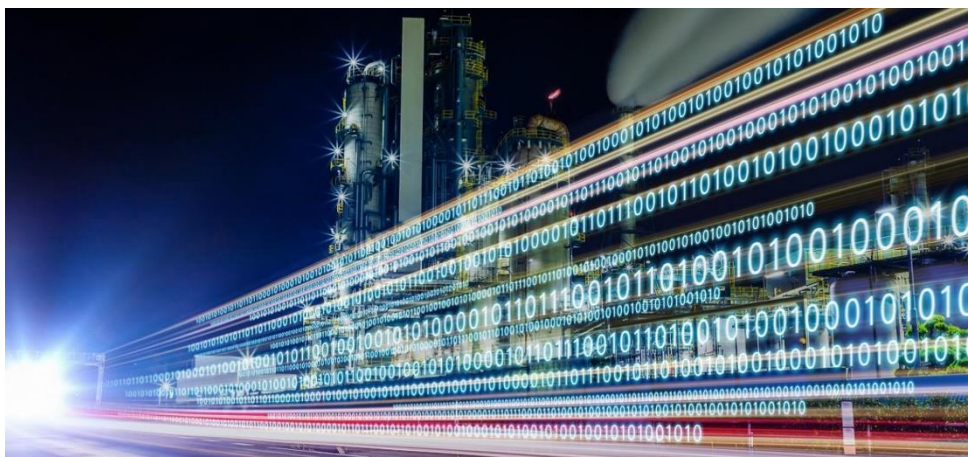
‘Allocate energy to construct a normalised data structure, yet run out of budget before building a viable presentation area based on dimensional models.’



I tend not to view this pitfall as being as much of an issue as the others because I've encountered transactional databases that have not been normalised enough to be scalable. When they do need normalising as a retrofit, they usually require extensive work and contingency workarounds so as not to disrupt the business when they are being upgraded. In the Kimball world, making changes to dimensional models poses less of a business risk than altering the source transactional databases. Enough resources should be set aside to build them but I would arguably state that the transactional databases running user applications should take priority. Seldom do small to mid-sized businesses invest in resources to create fully-conforming fact and dimension tables, let alone OLAP data cubes, but rather, end up building a working hybrid of normalised and Kimball-based structures.

Pitfall 6

'Pay more attention to backroom operational performance and ease of development than to front-room query performance and ease of use.'



Interestingly, my experience with this pitfall has often been the reverse by providing users easily readable data structures with good query performance but neglecting backroom operational performance by using substandard virtual servers, each hosting far too many databases and applications. However, I certainly agree that providing front-end users in the business with quick-loading, easily accessible data complete with training, online tutorials and documentation is extremely valuable.

Pitfall 5

‘Make the supposedly queryable data in the presentation area overly complex. Database developers who prefer a more complex presentation should spend a year supporting business users; they’d develop a much better appreciation for the need to seek simpler solutions.’



Generally, most business users seek simplicity in how data is presented. Moreover, access to this data should be limited to as few sources as possible. I've come across systems whereby users need to hunt down a dozen or more data sources each with its own ODBC connection and then combine in another application; for example, Microsoft Access. This is where the data warehouse should really excel; the ability to create an effective semantic model of what the data represents. I would like to add to this pitfall that it is very important to retain the lineage of the data; in other words, where it came from in the first place.

Pitfall 4

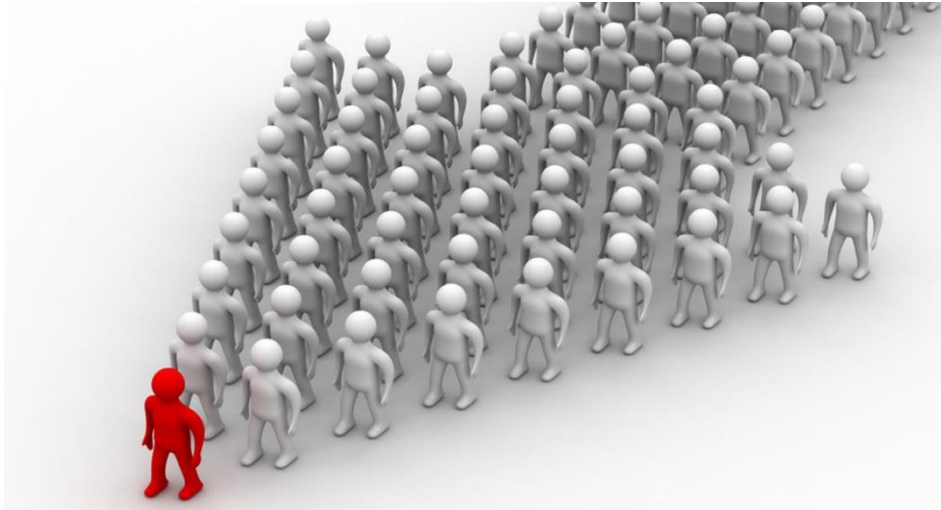
‘Populate dimensional models on a standalone basis without regard to a data architecture that ties them together using shared, conformed dimensions.’



The potential to generate lookup data which varies across data warehouse is likely to happen when dimensions are not consistent, unique and uniform across the data warehouse. For example, I’ve worked with data warehouses with dimensional structures which are virtually identical in nature but coming from different sources, administered under different owners and updated haphazardly. The classic example is the ‘time’ dimension; usually generated via a script sourced from the Internet. With inconsistent dimensional models, the end-user often has absolutely no idea why data from one report could vary significantly from a report generated elsewhere. One good practice is to tie in the data model with a Master Data Management (MDM) solution containing reference data which can be consumed universally throughout the data warehouse. If no official MDM solution is available, then the principles of using an MDM solution can still apply but using conventional dimensions provided there is sufficient governance and documentation in place.

Pitfall 3

‘Load only summarised data into the presentation area’s dimensional structures.’



In every ETL (extract, transform and load) process I've created between transactional databases and reporting data warehouses, I have never loaded data into the data warehouse pre-summarised. True, I might add additional columns with surrogate keys to identify data lineage or other metadata, but I've always ensured that data in its atomic format is available in the data warehouse should it be required. To achieve this, output views from the transactional databases are replicated as table structures in the data warehouse staging areas before being merged into target tables. If only summarised data is presented, determining data lineage and the ability to add or remove atomic data is often very difficult.

Pitfall 2

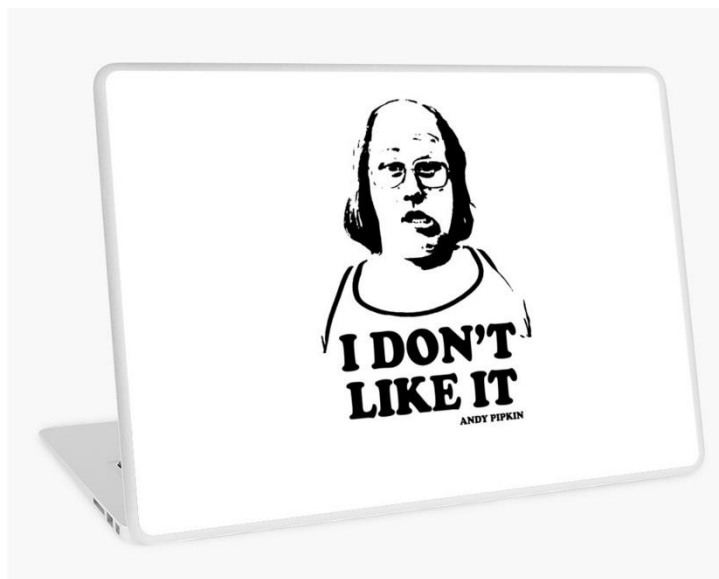
'Presume that the business, its requirements and analytics, and the underlying data and the supporting technology are static.'



I have never developed a data architecture that is limited to a business with static requirements and analytics. Furthermore, I have never come across a static business. Poor database modelling often involves developing repeated data structures along with hard-coded names and inconsistent naming conventions. Striking a balance of creating the right-size fact and dimension tables or the degree of normalisation is no small feat; however, exercising common sense to develop data structures which are scalable enough to adapt to the evolution of the business is important.

Pitfall 1

‘Neglect to acknowledge that data warehouse success is tied directly to user acceptance. If the users haven’t accepted the data warehouse as a foundation for improved decision making, then your efforts have been exercises in futility.’



I wholeheartedly agree with this fatal pitfall. If the users don't like it, they won't use it. It's game over.