The Lost Art of Dimensional Modeling

medium.com/sqldbm/the-lost-art-of-dimensional-modeling-13848db41637

Serge Gershkovich October 4, 2021

From business process to business warehouse, step by step.

With the advent of cloud data warehousing, the time and monetary costs of working with data have dropped considerably. Platforms like Snowflake's Data Cloud eliminate the cost barrier for new projects and offer features that allow for rapid prototyping like zero-copy cloning and scalable warehouses.

This drop in storage and processing costs means that design adjustments are more forgiving and less painful than they used to be. Many data engineers will skip the dimensional modeling phase altogether and jump straight into transformation while making adjustments along the way.

While this approach may be feasible in the early stages of a project, once data is loaded, and dependencies are built on top of core tables, changes to the data model can become costly to the point of being prohibitive.

Moor's Law does not trump the Pareto Principle — 20% invested in design at the start of a project will still save 80% on re-work and patches down the line, even as storage and computing costs continue to fall.

In short, there is no substitute for the fundamentals of dimensional modeling.

Prerequisites

Before diving into the process itself, it helps to set down some guidelines early on to help keep the project on track.

Best practices like establishing naming conventions for tables and column names and assigning stakeholders for project deliverables are well-known methods to reduce complexity and keep things running smoothly. But other considerations are not so obvious and are often overlooked.

One language

> And the LORD said, Behold, the people is one, and they have all one language; and this they begin to do: and now nothing will be restrained from them, which they have imagined to do.
>
> — Genesis 11:6 KJV

The Tower of Babel is a biblical origin myth explaining the world's numerous languages. But it can also be seen as a cautionary tale highlighting the importance of shared terminology to the success of a great endeavor.

Data modeling is mostly pattern-based — there is rarely a need to "reinvent the wheel." This is why, for both clarity and economy of communication, it is essential to know the standard terms that from its building blocks.

This applies to the fundamentals (e.g., facts, dimensions, measures,) more advanced concepts (e.g., type II SCDs, upserts, degenerate dimensions), and any team-specific conventions that may be established during the life of the project.

For a periodic refresher of core terms and concepts, keep a copy of Ralph Kimball's iconic Data Warehouse Toolkit handy (an abridged overview of terms can also be found on his website.)

One source of truth

Dimensional modeling, as we'll see in detail below, is a collaborative and iterative process. It requires that diverse profiles across the organization work together to arrive at a suitable solution (i.e., it satisfies business requirements and is efficient and maintainable long term.)

With so many stakeholders involved in shaping its design, the solution is guaranteed to undergo many changes before finalization. For this reason, it's essential to use a tool that can support the collaborative process through the entire lifecycle — enabling real-time collaboration, dynamic sharing, tracking, and avoiding stale documents and knowledge silos.

For this article, I will use SqlDBM online modeling tool to follow along with the exercise. SqlDBM is versatile enough to allow for high-level whiteboard-style design to start. Unlike dedicated diagramming tools (e.g., Lucidchart, Vizio), SqlDBM generates neatly formatted, database-specific DDL as the model evolves, thus avoiding re-work.

With SqlDBM, the map becomes the territory.

The Process

In this article, we will use the sales process of a retail business and walk through the various stages of dimensional modeling.

1. Select the business process

The first step is selecting a (single) business process to model. This is not as obvious as it sounds and thus, bears mention. Never attempt to architect the entire warehouse at once — build it up one business process at a time.

Choosing a business process is a business-driven decision and will need to be prioritized against alternate business processes based on relevance and necessity. The BI team can help in the estimation, but the company's needs should determine which business process to focus on.

During this process, the BI team should meet with subject matter experts from the business and perform high-level data profiling. This will help form a high-level technical proposal which can then be used for budget estimates and timelines.

2. Determine the grain

Once the business process has been selected, the next step is to determine (minimum) the grain at which it will be analyzed. This will again be a business decision driven by existing company needs.

If we are looking at an online retailer, we have very detailed order data from varied sources. We can, for example, determine the device id and operating system of every order and track it down to the millisecond of placement. This may be relevant for marketing or customer segmentation but will not add much value to overall sales tracking.

If the business users decide that orders should be analyzed daily by product type, then "daily by product type" becomes the minimum grain. The grainestablishes what a single fact table row represents. Henceforth this should be treated as a binding agreement with the stakeholders because deviating from it demands considerable design changes.

Remember that regardless of the decided grain, always extract source data at the lowest and fullest detail available in the source system (i,e., follow the extract, load, transform (ELT) pattern.) If today's requirements call for "daily by product type" reporting, there is no guarantee that they won't evolve to "hourly by product id" tomorrow.

3. Identify the dimensions

In the previous step, we identified several of the primary dimensions for business analysis. Now it's time to discover the rest.

As part of the business process selection activity, common dimensions and alternate business processes may have been uncovered. This forms the basis for the dimensional business matrix.

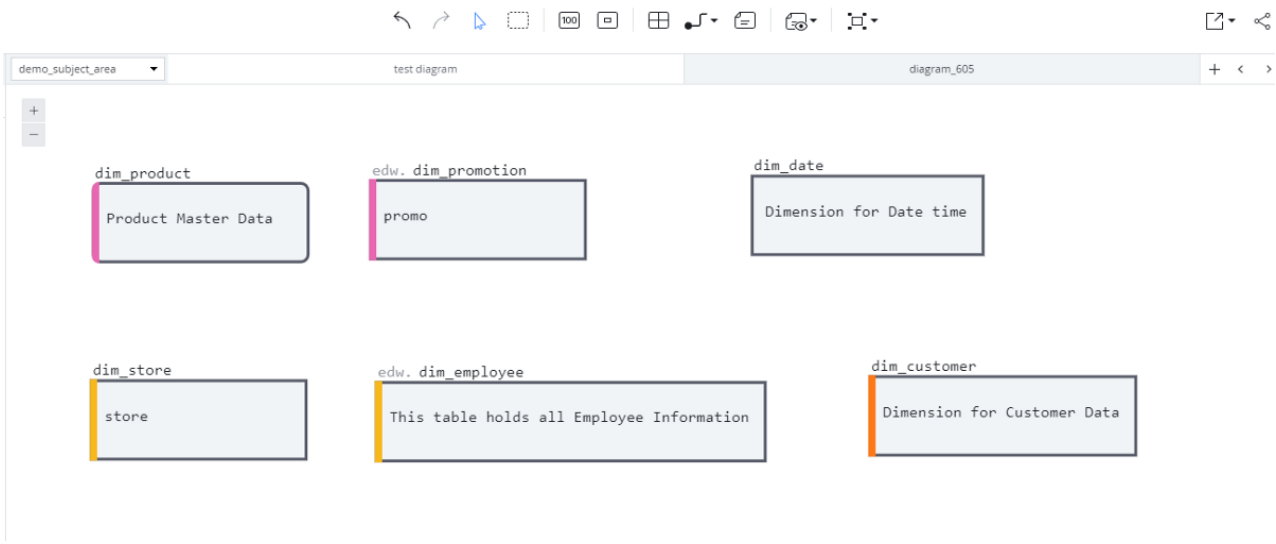
Even if we are only focused on one business process (retail sales), a dimensional matrix is a helpful tool for identifying correlated business processes by highlighting their common dimensions.

	COMMON DIMENSIONS						
BUSINESS PROCESSES	Date	Product	Warehouse	Store	Promotion	Customer	Employee
Issue Purchase Orders	X	X	X				
Receive Warehouse Deliveries	X	X	X				X
Warehouse Inventory	X	X	X				
Receive Store Deliveries	X	X	X	X			X
Store Inventory	X	X		X			
Retail Sales	X	X		X	X	X	X
Retail Sales Forecast	X	X		X			
Retail Promotion Tracking	X	X		X	X		
Customer Returns	X	X		X	X	X	X
Returns to Vendor	X	X		X			X
Frequent Shopper Sign-Ups	X			X		X	X

sample dimensional matrix from

A dimension is typically expressed as a noun (e.g., store, employee, vehicle.) A noun will naturally have attributes (e.g., name, description, address), which should be modeled to add richness and context to the reports.

Having identified the existing dimensions, we can begin their basic whiteboarding in preparation for the details that will be worked out in subsequent steps.



Identifying dimensions in SqlDBM through logical modeling

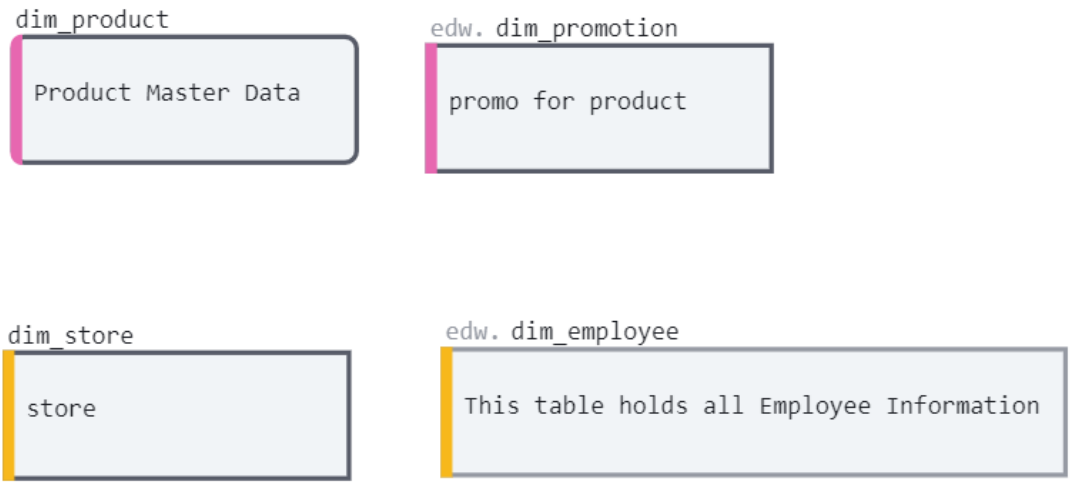
4. Identify the dimension relationships

Now that the dimensions have been identified, we need to understand the nuances of how they relate to one another.

Are promotions applied at store or product level (or both)? Is an employee tied to a specific store, or can they be assigned to multiple?

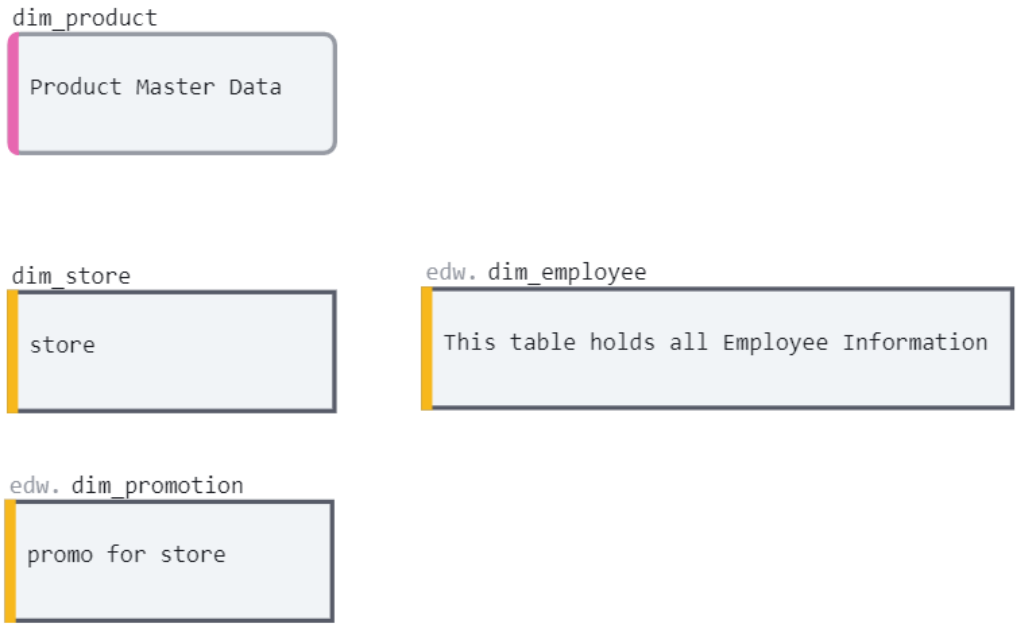
Although these questions will drive the technical design, they are functional questions that only the business experts can answer. Make sure the business experts stay engaged throughout the modeling process and resolve functional doubts as they arise.

So, are promotions applied at product level?



promo at product level

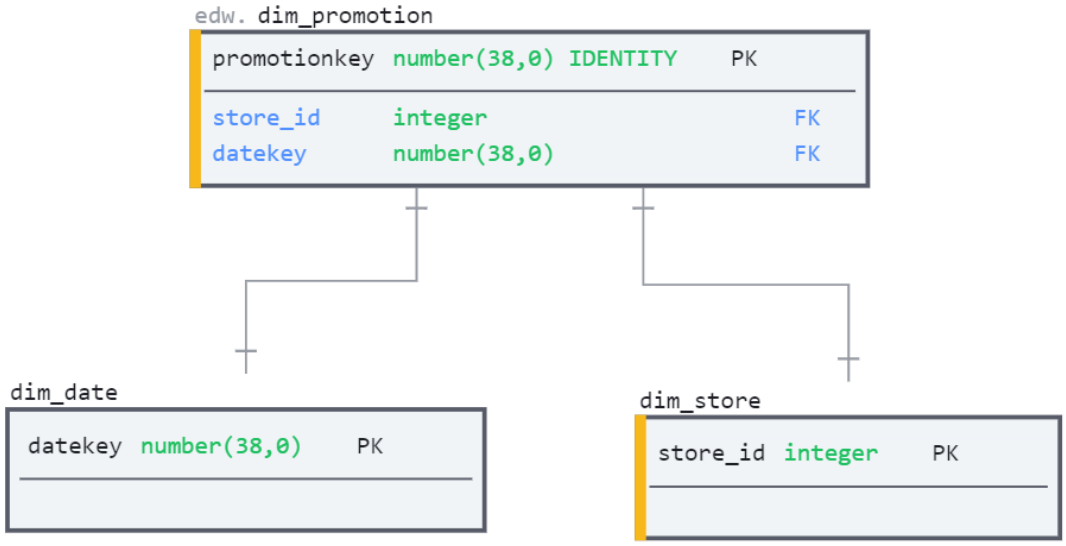
Or does a promotion apply to the entire store?



promo at store level

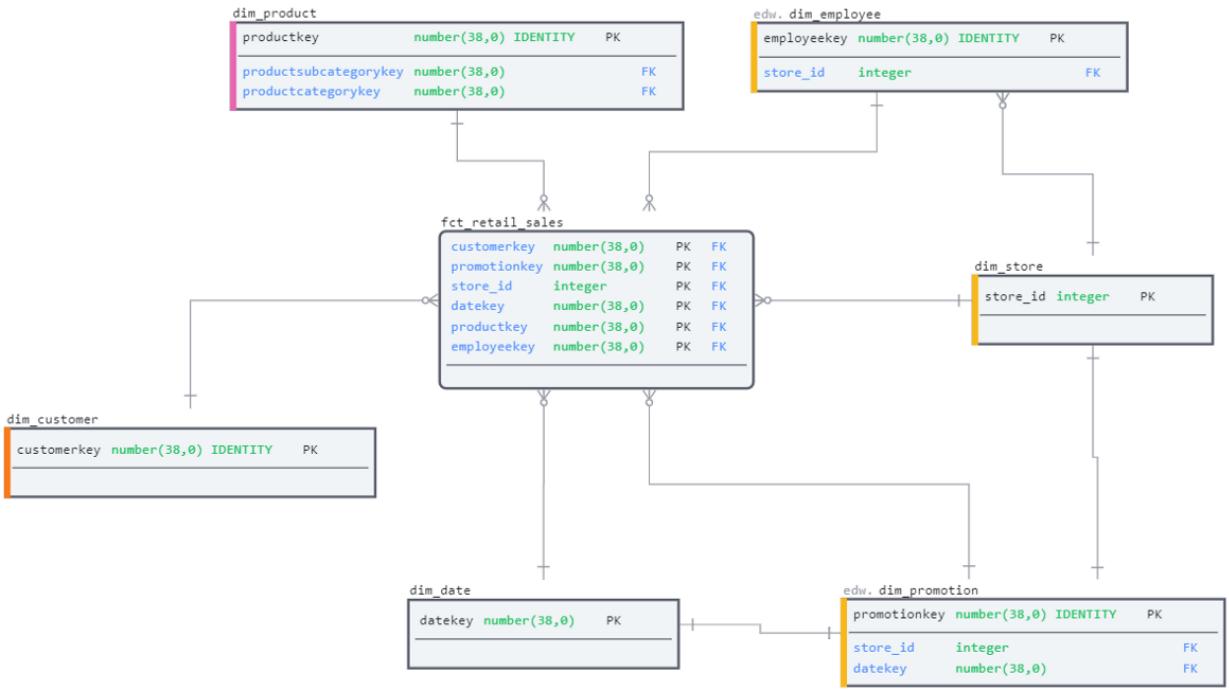
Logical modeling allows for rapid prototyping before moving on to physical design. This simplified view allows for non-technical team members to take part in the discussion.

For this example, let us go with the latter option, of holding promotions at store level.



5. Identify the facts

Combining the previously identified dimensions (*what, when*) with quantitative measures (how **much, many**) will yield facts. For example, customer *A* bought **X** of product *B* at a price of **Y** dollars from employee *D* at store *E* using promotion *F* on date *G*. This combination of dimensions and measures give us the fact table for the retail sales business process.



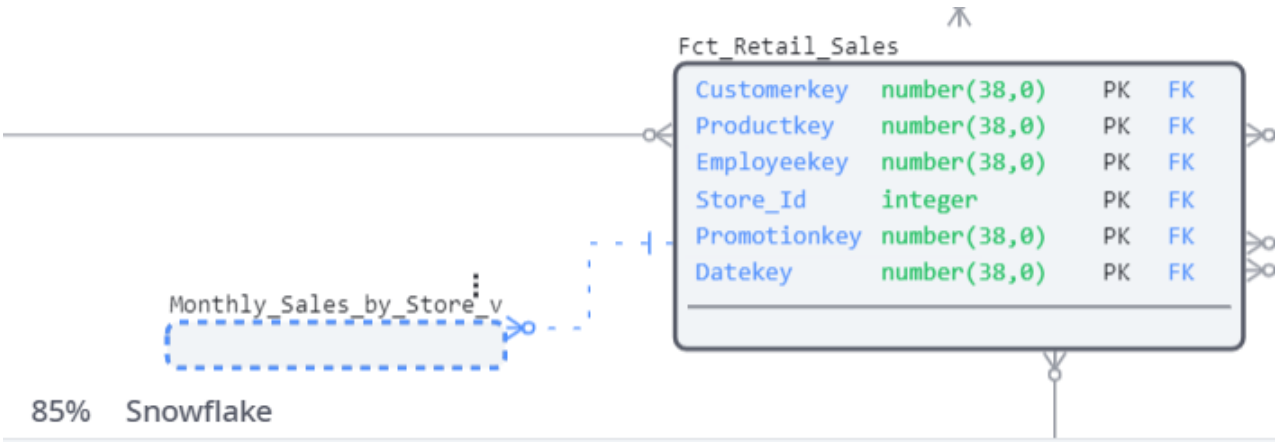
the fact table of the retail sales business process with associated dimensions

Using this fact table, we can answer any business question regarding retail sales through a combination of filtering and aggregation. The diagram can now be expanded to show relevant technical details like primary and foreign keys.

For example, if the business wants to know the monthly sales by store for product B, we can filter and sum over the fact table accordingly:

```
SELECT store_id, date_trunc('MONTH', datekey) as sale_month, sum(price) as monthly_sales_usdFROM fct_retail_sales WHERE productkey = 'B' GROUP BY store_id, datekey
```

This can also be done in view as part of the project itself:



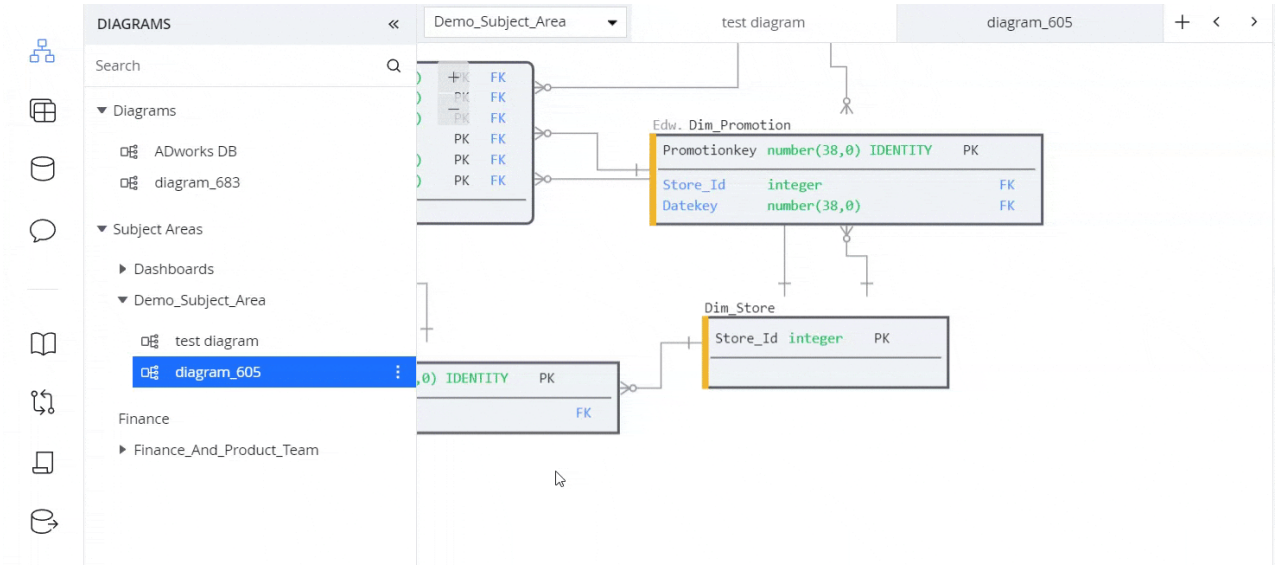
SQL script

```
Create View Monthly_Sales_by_Store_v

  Comment = 'for monthly reporting'

AS
(
SELECT
store_id,
date_trunc('MONTH', datekey) as sale_month,
sum(price) as monthly_sales_usd
FROM fct_retail_sales
WHERE productkey = 1
GROUP BY 1,2

);
```

Notice that having declared a very granular level of detail as our basic fact allows us to aggregate up to a simpler level of detail. Had our fact table been declared at month level, there would be no way to drill down and get daily details.

6. Deploy

Now that the facts, measures, grain, and relationships have been agreed, the time has finally come to access the database and create the physical objects. Here, it's essential to keep track of all the technical and functional details that have been agreed upon throughout the modeling exercise (e.g., column lengths, data types, and table properties.)



If you've been using SqlDBM to follow along, then you'll be able to forward engineer the required DDL directly from the diagram(the map becomes the territory, remember?) Using Excel, you could also create formulas that generate SQL based on the details you've entered. Whatever solution you decide on, make sure that the design always stays in sync with the technical details.

Conclusion

Having walked through the steps of the dimensional modeling process, we've seen that it is an iterative and collaborative endeavor in which requirements and design decisions are almost guaranteed to change. To keep the project running smoothly, it is vital to use tools and methods which can easily absorb such design shocks while keeping everyone in sync.

Once tools and methods have been established, it is a repeatable, pattern-based exercise. Choose these wisely — then plan, model, repeat.