snowflake®

5 BEST PRACTICES FOR DATA WAREHOUSE DEVELOPMENT

EBOOK

TABLE OF CONTENTS

INTRODUCTION

Cloud technology has revolutionized how businesses store, access, and analyze data. Whether your organization is creating a new data platform from scratch or re-engineering a legacy data warehouse system to take advantage of new capabilities, a handful of guidelines and best practices will help ensure your project's success. Some of those best practices may seem obvious, but all too often, businesses fail to spend time up front setting and documenting these decision points, resulting in headaches and inefficiency down the road.

In this ebook, we outline five recommendations for putting structure around your data strategy and getting alignment across your business, so the data warehouse you create meets both current and future business needs. These best practices for data warehouse development will increase the chance that all business stakeholders will derive greater value from the data warehouse you create, as well as lay the groundwork for a broader enterprise data platform that can grow and adapt as your business needs change.

1. CREATE A DATA MODEL

The first key step in any data program is to create a data model: an abstract representation that organizes elements of data and describes how they relate to one another and to properties of their real-world entities. A data model establishes a common understanding and definition of what information is important to the business, as well as the business's overall data landscape. Having a data model gives you a way to document the data sets that will be incorporated into the data warehouse, the relationship between those data sets, and the business requirements that the platform seeks to fulfill.

Could you create a data warehouse without a data model? Yes, but when you decide not to take this basic step, you lose many valuable insights. Creating a comprehensive data model is often an eye-opening exercise for businesses, as it forces different functional teams to agree on the definition and delineation of data assets and business requirements of the data warehouse before development is underway.

A well-defined data model drives a positive impact long after the data warehouse (or data mart) is live. For example, a data model establishes data lineage for all the objects in the data warehouse, making it easier to onboard new team members or to bring new data objects into the data warehouse as business needs change.

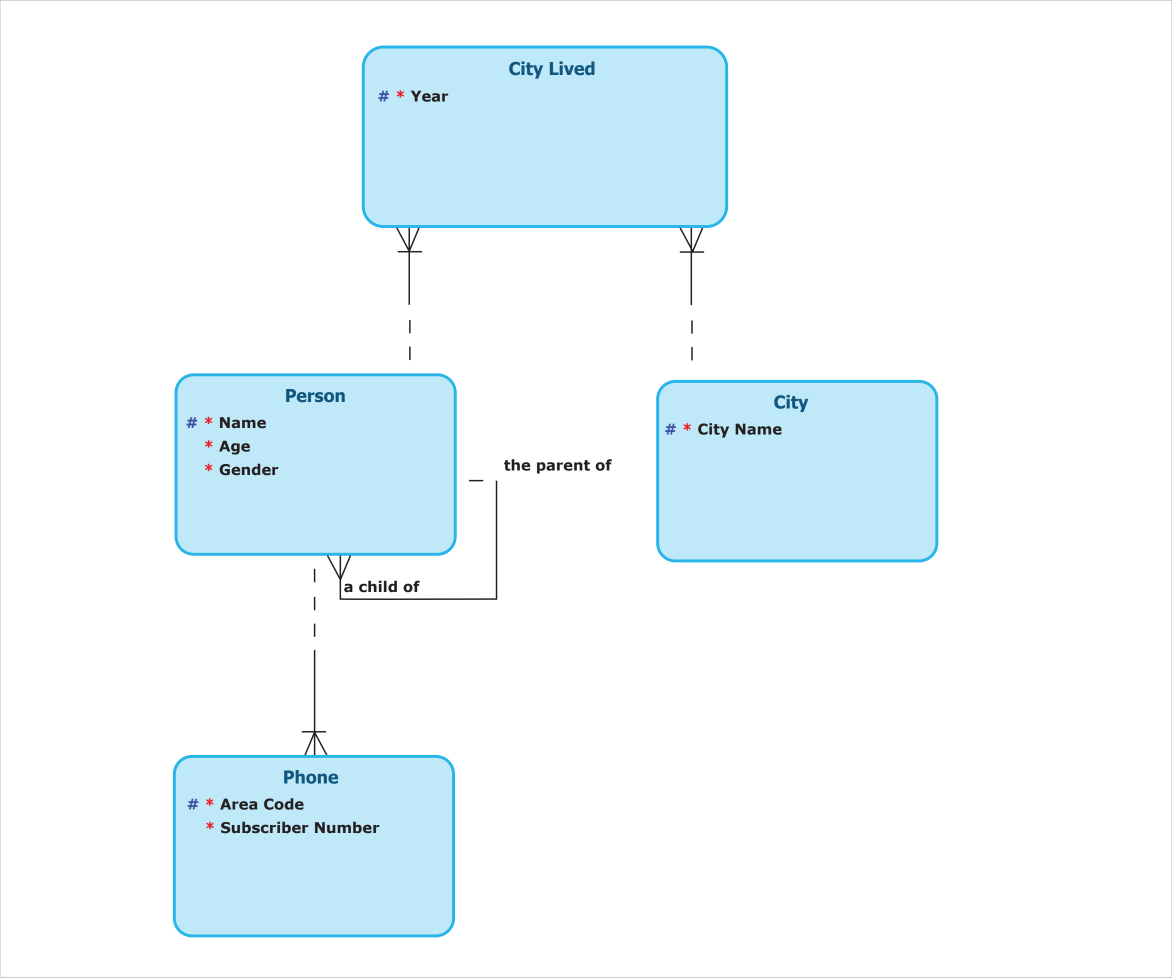

Figure 1: A typical 3NF (third normal form) logical data model

The data model also provides clear documentation of the content, context, and sources. This makes it easier to audit or to comply with new data requirements, such as those presented by GDPR (the EU's General Data Protection Regulation framework that sets guidelines for the collection and processing of personal information from individuals).

Having a strong data model also helps prevent confusion and costly reengineering down the road. It's always a good idea to incorporate a source-agnostic integration layer that enables analysis across multiple data sets based on the data sets' commonalities.

A data warehouse brings together many different sources and types of data, including traditional data sets such as customer relationship management (CRM) data and enterprise resource planning (ERP) data, as well as data sets like blogs, Twitter feeds, IoT data, and even data sets that have yet to be invented. This is why having a flexible integration layer that isn't too tightly tied to any single system will help future-proof your data warehouse.

A highly effective data model should employ definitions and semantic structures that are defined by the business domain, not by the specific definitions of any single source system. For example, one CRM system may refer to customers as "cust," while another refers to "cust_ID." Establishing a business-wide semantic rule for how users should name, access, and analyze that data across data sets is key to the data warehouse's success.

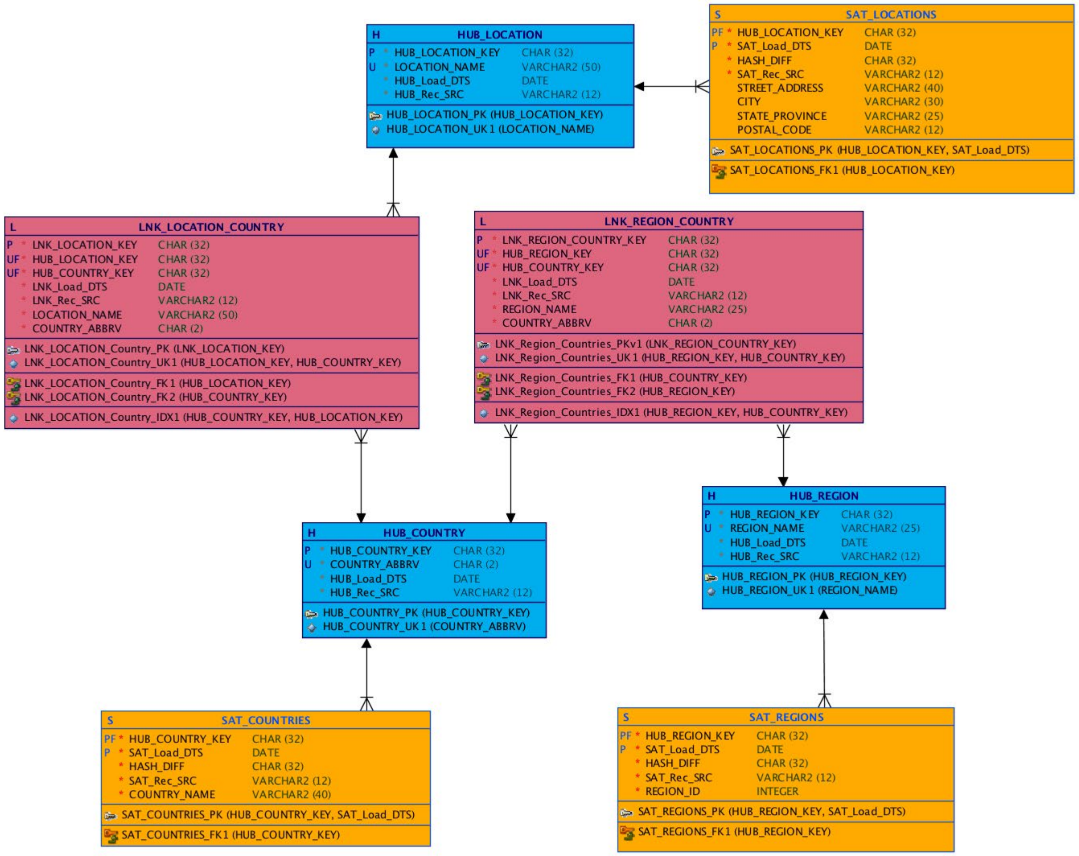
H HUB_LOCATION
P * HUB_LOCATION_KEY CHAR (32)
U * LOCATION_NAME VARCHAR2 (50)
* HUB_Load_DTS DATE
* HUB_Rec_SRC VARCHAR2 (12)
HUB_LOCATION_PK (HUB_LOCATION_KEY)
HUB_LOCATION_UK1 (LOCATION_NAME)
S SAT_LOCATIONS
PF * HUB_LOCATION_KEY CHAR (32)
P * SAT_Load_DTS DATE
* HASH_DIFF CHAR (32)
* SAT_Rec_SRC VARCHAR2 (12)
STREET_ADDRESS VARCHAR2 (40)
CITY VARCHAR2 (30)
STATE_PROVINCE VARCHAR2 (25)
POSTAL_CODE VARCHAR2 (12)
SAT_LOCATIONS_PK (HUB_LOCATION_KEY, SAT_Load_DTS)
SAT_LOCATIONS_FK1 (HUB_LOCATION_KEY)
L LNK_LOCATION_COUNTRY
P * LNK_LOCATION_KEY CHAR (32)
UF * HUB_LOCATION_KEY CHAR (32)
UF * HUB_COUNTRY_KEY CHAR (32)
* LNK_Load_DTS DATE
* LNK_Rec_SRC VARCHAR2 (12)
* LOCATION_NAME VARCHAR2 (50)
* COUNTRY_ABBRV CHAR (2)
LNK_LOCATION_Country_PK (LNK_LOCATION_KEY)
LNK_LOCATION_Country_UK1 (HUB_LOCATION_KEY, HUB_COUNTRY_KEY)
LNK_LOCATION_Country_FK1 (HUB_LOCATION_KEY)
LNK_LOCATION_Country_FK2 (HUB_COUNTRY_KEY)
LNK_LOCATION_Country_IDX1 (HUB_COUNTRY_KEY, HUB_LOCATION_KEY)
L LNK_REGION_COUNTRY
P * LNK_REGION_COUNTRY_KEY CHAR (32)
UF * HUB_REGION_KEY CHAR (32)
UF * HUB_COUNTRY_KEY CHAR (32)
* LNK_Load_DTS DATE
* LNK_Rec_SRC VARCHAR2 (12)
* REGION_NAME VARCHAR2 (25)
* COUNTRY_ABBRV CHAR (2)
LNK_Region_Countries_PKv1 (LNK_REGION_COUNTRY_KEY)
LNK_Region_Countries_UK1 (HUB_REGION_KEY, HUB_COUNTRY_KEY)
LNK_Region_Countries_FK1 (HUB_COUNTRY_KEY)
LNK_Region_Countries_FK2 (HUB_REGION_KEY)
LNK_Region_Countries_IDX1 (HUB_REGION_KEY, HUB_COUNTRY_KEY)
H HUB_REGION
P * HUB_REGION_KEY CHAR (32)
U * REGION_NAME VARCHAR2 (25)
* HUB_Load_DTS DATE
* HUB_Rec_SRC VARCHAR2 (12)
HUB_REGION_PK (HUB_REGION_KEY)
HUB_REGION_UK1 (REGION_NAME)
H HUB_COUNTRY
P * HUB_COUNTRY_KEY CHAR (32)
U * COUNTRY_ABBRV CHAR (2)
* HUB_Load_DTS DATE
* HUB_Rec_SRC VARCHAR2 (12)
HUB_COUNTRY_PK (HUB_COUNTRY_KEY)
HUB_COUNTRY_UK1 (COUNTRY_ABBRV)
S SAT_COUNTRIES
PF * HUB_COUNTRY_KEY CHAR (32)
P * SAT_Load_DTS DATE
* HASH_DIFF CHAR (32)
* SAT_Rec_SRC VARCHAR2 (12)
* COUNTRY_NAME VARCHAR2 (40)
SAT_COUNTRIES_PK (HUB_COUNTRY_KEY, SAT_Load_DTS)
SAT_COUNTRIES_FK1 (HUB_COUNTRY_KEY)
S SAT_REGIONS
PF * HUB_REGION_KEY CHAR (32)
P * SAT_Load_DTS DATE
* HASH_DIFF CHAR (32)
* SAT_Rec_SRC VARCHAR2 (12)
* REGION_ID INTEGER
SAT_REGIONS_PK (HUB_REGION_KEY, SAT_Load_DTS)
SAT_REGIONS_FK1 (HUB_REGION_KEY)

Figure 2: An example data model using the Data Vault modeling approach

As a company goes through changes, mergers, and acquisitions, the CRM system it is using today may be replaced by a different CRM system. If your data model is tightly coupled to a specific source system, then you will have to do a lot of reengineering to integrate the second source system that replaces the legacy system. A source-agnostic integration layer makes data mapping much easier, so you can swap an old source system for a new source system without affecting downstream reports or having to change user behavior.

Within the data model, it's critical to select a standard approach. The main types of data modeling standards used in data warehouse design include:

- **3NF:** 3NF, which stands for "third normal form," is an architectural standard designed to reduce the duplication of data and ensure referential integrity of the database.[1]
- **Star schema:** The simplest and most widely used architecture to develop data warehouses and dimensional data marts, the star schema consists of one or more fact tables referencing any number of dimension tables.[2]
- **Data Vault (DV):** Developed specifically to address agility, flexibility, and scalability issues found in other approaches, DV modeling was created as a granular, non-volatile, auditable, easily extensible, historical repository of enterprise data. It is highly normalized and combines elements of 3NF and star models.[3]

Each architecture has its advantages, but the choice of which to adopt will depend on the business needs of the organization.

Frankly, more important than which architecture your organization selects is that it selects, documents, and continually supports this architecture as part of developing a data model for the warehouse. Doing so will enable future efficiency, allowing for a single support and troubleshooting methodology that will make it easier for new team members to ramp up more quickly.

2. ADOPT AN AGILE DATA WAREHOUSE METHODOLOGY

In the past, data warehouse (or even data mart) creation was a large, monolithic, multi-quarter, or multi-year effort, subject to the traditional "waterfall" process. In the modern age, that's no longer the norm as many organizations are choosing to adopt a more flexible and iterative, or Agile, design approach.

With business needs changing faster than ever, and new data sources coming online more quickly, businesses need to be able to adapt and leverage these inputs concisely and rapidly. That means learning to build data and analytic solutions in an incremental and Agile fashion. With proper planning that aligns to a single source-agnostic integration layer, large data projects can now be broken down into smaller pieces that can be delivered more frequently, thus providing incremental business value much more quickly.

Data warehousing architects are adopting the Agile methodology, which first appeared in the software development world, to achieve this goal. In the Agile methodology, requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and customers. When applied to data warehouse conception and construction, the Agile methodology enables businesses to activate new data sets and solve new business challenges more quickly.[4]

Within the Agile worldview, a variety of approaches have emerged to help deliver value faster, including:

- **Scrum—**Named for the rugby formation in which forwards interlock arms and advance, Scrum is the most widely used process framework for Agile development. A lightweight framework, Scrum emphasizes daily communication and the flexible reassessment of plans that are carried out in short, iterative phases of work.[5] Ralph Hughes codified Scrum's application to data warehousing in a series of seminal works that are useful to businesses adopting this approach.
- **Kanban—**Kanban is a method for managing the creation of products with an emphasis on continuous delivery without overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively. Named for the "Kanban" cards that track production in a factory, Kanban was created by Taiichi Ohno, an industrial engineer at Toyota, to improve manufacturing efficiency.
- **BEAM—**BEAM, or Business Event Analysis and Modelling, was introduced by Lawrence Corr and Jim Stagnitto in their groundbreaking work, Agile Data Warehouse Design. BEAM focuses on business events, rather than on known reporting requirements, to model the whole business process area. It leverages seven dimensional types (the seven Ws: who, what, when, where, how, how many, and why) to identify and then elaborate on business events.[6]

To leverage the benefits of Agile development more fully, an Agile data platform is very helpful. Cloud-based data platforms provide that structural flexibility and elasticity, enabling rapid scaling as business needs evolve. Cloud-based data platforms require less effort, maintenance, and administration to be useful, and they can grow and adapt to changing business requirements. By leveraging a modern cloud service, teams can spend less time tuning queries and provisioning storage and more time addressing immediate business challenges and delivering business value.

Leveraging Agile methodologies and structures is no small undertaking. It requires a cultural commitment within the organization and is often a significant shift in mindset and workflow from traditional data warehousing workflows. Retooling an IT team to work comfortably in an Agile environment can take six to 12 months, which may seem paradoxical given the Agile methodology's goal of delivering value more quickly. This transition can be accelerated by engaging with a seasoned Agile coach. But once the shift is made, teams can begin to deliver new incremental changes to the data warehouse in weeks, instead of months.

3. FAVOR ELT OVER ETL

In the past, data warehousing development took an extract-transform-load (ETL) approach, extracting the data to be imported into the data warehouse from the source systems, cleaning it or applying business rules to it on an external server, and then loading it into the target data warehouse. Increased data platform computing power and capabilities have yielded a new preferred approach: extract-load-transform (ELT).

In the ELT approach, raw data is extracted from the source and loaded, relatively unchanged, into the staging area of the data warehouse. Metadata, load date, or source information may be added to the data, and then it is brought directly into the data warehouse. Once inside the data warehouse, businesses can use the power of the database to perform transformations, whether that's changing the structure of the data (that is, applying a data model), applying business rules, or performing data quality measures to cleanse the data (for example, correcting incomplete addresses, standardizing data field names, and resolving duplicates).

The ELT approach has two distinct advantages: cost savings and greater traceability. ELT helps realize cost savings as it allows businesses to leverage the power of the data platform to transform data, instead of using an external server. Cloud-based computing power is typically much less expensive than performing transformations and data manipulation on an external server, so moving data to the cloud directly is faster and cheaper. The ELT approach also makes it easier to audit and trace the data in the future, because it provides an image of the original source data directly within the data platform. In this way, the data warehouse itself can play the role of what has come to be known as a "data lake," where raw data is stored persistently.

4. ADOPT AN AUTOMATION TOOL

The goal of the data warehouse is to activate and deliver data more quickly so it can inform business decisions and drive greater value. One way to increase speed of delivery is to adopt the Agile methodology. Another is to adopt automation tools that can help develop and deploy code more quickly. Because many data warehouse methodologies are pattern-based, the coding required for loading and structuring data is often repeatable, which means it can be automated. A number of tools on the market automate some or even all of the design and build tasks, and the list grows daily.

Automation allows businesses to leverage their resources more fully, iterate faster, and enforce coding standards more easily. It enables the creation of standardized code, which is incredibly useful in organizations where the ETL code and data models were traditionally developed by hand. Automation provides a documented standard for these different artifacts, as well as an enforcement and quality assurance (QA) mechanism to monitor that all developers and designers are following that standard.

Automation tools that use templates to generate code are especially helpful, because they enforce standards by making them the preferred properties within the templates themselves. This makes onboarding faster, as new developers and designers will use these standard tools, guaranteeing consistent implementation and shortening the learning curve. A consistent implementation has the added benefit of being easier to test and debug because code is developed using the same standards.

Iteration also becomes faster by using these tools, because automated code generators tend to not make syntax errors. Updating code typically means adding a new object to the tool or changing the templates' properties at the global level, generating new code that is immediately available for deployment in the environment for testing and validation.

5. TRAIN YOUR STAFF ON NEW APPROACHES

A move to the Agile methodology or automated code development isn't just a shift in skill sets—it's a shift in mindset. Training and education are required to ensure the team is leveraging these new approaches and technologies effectively. This may mean bringing in external experts to train teams on the Scrum best practices or educating teams on the benefits, rules, and best practices for whatever standard architecture the business has adopted for its data platform.

Many industry resources are available to help manage the transition to the Agile methodology. The **Agile Alliance**, a global nonprofit member organization dedicated to promoting the concepts of Agile software development as outlined in the Agile Manifesto, offers many training options for introducing Agile concepts. The **Scrum Alliance** offers certifications and training for foundational and advanced Scrum training. Likewise, Data Vault bootcamp and certification is offered by selected partners through the **Data Vault Alliance**.

As with any new process and cultural change, organizations should manage the adoption curve to ensure a consistent and effective shift to the new approach in day-to-day operations. Identifying pilot or proof-of-concept projects for initiating the teams to the new approaches will ensure practitioners build and master the skills in protected, yet real, scenarios that will accelerate competence and abilities in these new skills.

SUMMARY

All of the best practices outlined in this ebook require an upfront investment to achieve the long-term business value they can deliver. But, the return on that investment is twofold: It will lay the foundation for a successful data analytics program at the outset and accelerate the successful delivery of incremental business value to your data environment long after the first production release.

As business requirements change and the desire to gain more value from even more data and data types continues to accelerate, having these best practices in place will allow you to think and grow beyond the traditional data warehousing use cases. With a solid foundation and agile platform, you will be able to expand into new data realms and meet new demands by broadening the program to support data science, machine learning, AI, and maybe even data monetization. With today's flexible and scalable cloud resources, there really is no limit to what you can achieve with your data.

snowflake®

ABOUT SNOWFLAKE

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, and execute diverse analytic workloads. Wherever data or users live, Snowflake delivers a single data experience that spans multiple clouds and geographies. Thousands of customers across many industries, including 543 of the 2022 Forbes Global 2000 (G2K) as of October 31, 2022, use Snowflake Data Cloud to power their businesses.

Learn more at **snowflake.com**

CITATIONS

[1] en.wikipedia.org/wiki/Third_normal_form

[2] en.wikipedia.org/wiki/Star_schema

[3] snowflake.com/blog/data-vault-modeling-and-snowflake

[4] Agiledata.org/essays/dataWarehousingBestPractices.html

[5] scrum.org/resources/what-is-scrum

[6] bisystembuilders.com/beam