SAP to Snowflake: Moving Data Up

By David Rice, Jan van Ansem, Frank Khan Sullivan and Tom Bruce

Edition 1

Contents

Chapter 1: Introduction

In today's data-driven world, organisations are relentlessly pursuing ways to improve their data management and analytics capabilities. Amidst the overwhelming flow of information, these organisations stand at the crossroads of innovation and obsolescence.

With every byte of data lies an opportunity: to gain a competitive edge, to unearth novel insights, or to redefine entire industries. But harnessing this power is no small feat. The journey from raw data to actionable intelligence is strewn with challenges, from the technical intricacies of data storage to the complexities of analysis. And at the heart of this journey is the complex nature of SAP, its underlying data structures and the challenges to extract, load and transform in Snowflake.

A Brief History of Data Warehousing on SAP

In the latter half of the 20th century, as businesses expanded and their operations grew complex, the need for effective data management systems became paramount. Data warehousing, a concept introduced in the late 1970s, emerged as a solution to address this need. By the 1990s, these warehouses became an integral part of enterprise IT infrastructure, paving the way for companies like SAP to profoundly influence the landscape.

The initial data warehousing concepts, put forth by pioneers like Bill Inmon and Ralph Kimball, set the stage for what would become a multi-billion-dollar industry. SAP, initially a player in the enterprise resource planning (ERP) space, recognized the potential of this trend early on. With its Business Information Warehouse offering introduced in 1998, SAP began its journey into the world of data warehousing.

In 2004, with the release of SAP NetWeaver, Business Information Warehouse became SAP Business Warehouse (SAP BW). The product received another significant update in 2013 when SAP customers could configure SAP BW to run on SAP HANA, an in-memory database. This offered several benefits, such as a simpler infrastructure and increased performance.

SAP BW/4HANA was released as a new product in 2016 and was a re-architected and optimised evolution of BW tailored specifically for HANA. In 2019, SAP announced SAP Data Warehouse Cloud, now called SAP Datasphere, which offered SAP HANA as a database as a service.

The Rise of Cloud Data Warehousing

In recent years, there has been a growing trend towards cloud data warehousing. This is due to several factors, including the increasing availability of cloud computing resources, the falling cost of cloud storage, and the growing demand for agility and scalability.

Cloud-based data warehouses, in contrast to their traditional on-premises counterparts, offer superior scalability, allowing organisations to seamlessly adjust capacity without the burden of procuring new hardware or software. This flexibility typically translates to cost-effectiveness, as expenses are tied only to the resources consumed. Furthermore, these modern warehouses often surpass traditional ones in security, given that providers top priority is always security.

At the forefront of this rising trend has been Snowflake, the leader in cloud data warehousing and now supporting non-traditional use-cases with the Snowflake Data Cloud.

Snowflake is a popular alternative for organisations that want to use SAP data to drive business decisions and perform advanced analytics. Let's examine why.

Chapter 2: Why SAP Customers are Choosing Snowflake

While SAP products, especially its data warehousing solutions, have been dominant forces in the enterprise software space for decades, there's a growing trend of businesses exploring alternatives. One of the top contenders that have emerged in recent years is Snowflake. This chapter will dive into the pain points experienced by SAP users and explore why Snowflake is increasingly being chosen as the remedy.

Challenges Faced by SAP Customers

While SAP data warehouse solutions many advantages, there remain notable challenges inherent to its architecture and the broader SAP ecosystem:

- **Rigid Design**: One of the often-cited criticisms of SAP products is its rigidity. Unlike more flexible systems, making changes to the data model, or adjusting to evolving business needs can be cumbersome in SAP. This rigidity often results in extended timelines for implementing changes or incorporating new data sources.

- **Integration with Third-Party Software**: SAP's ecosystem is, in many ways, a walled garden. Integrating with third-party applications, be it data sources, ETL tools, or other analytics platforms, is not always straightforward. While there were connectors and integration methodologies available, they were often not as seamless or efficient as users would prefer.

- **Cost Implications**: One of the major hurdles with SAP is the total cost of ownership. SAP's pricing structure, particularly for its data warehousing solutions, can be a significant point of contention for many enterprises. The high licensing costs, coupled with the expenditure for infrastructure and maintenance, make it an expensive proposition for many companies, especially when compared to cloud-based solutions that adopt a more scalable pricing model. It’s not just the licensing, but the esoteric nature of SAP systems means businesses often require specialist skill sets to manage, optimise, and troubleshoot the system. These SAP specialists are in high demand, which further increases operational costs.

Snowflake: Addressing the Challenges

Snowflake's Data Cloud has been able to address these challenges through a combination of architectural design and rich feature set.

1. **Shared Storage and Shared-Nothing Benefits**

This combination means that data is stored centrally (shared storage) but can be computed and processed by multiple, independent computational units (shared-nothing). This structure ensures that data is accessible and consistent across the platform while simultaneously allowing for high concurrency and performance.

2. **Flexibility with Semi-Structured Data**

Traditional systems often struggle or require complex processes to handle semi-structured data. Snowflake shines here. It natively supports semi-structured data types like JSON, Avro, and Parquet. This support means businesses can ingest, store, and directly query semi-structured data without cumbersome transformations, providing an agile response to varied data sources.

3. **Automated and On-the-Fly Scaling**

Snowflake's Virtual Warehouses can automatically scale, vertically or horizontally, based on workload. This feature ensures optimal performance even with varied workloads, without manual intervention.

The on-demand scaling aligns costs with actual usage, ensuring that businesses only pay for the computational power they consume.

4. **Broad Third-Party Ecosystem**

One of Snowflake's philosophical differences to SAP is its commitment to extensive integration. This gives SAP customers wider choice for best of breed ELT tools such as Matillion and Fivetran.

5. **Data Sharing**

SAP customers can now easily share data with their suppliers or customers through data sharing, more easily collaborate with different business units via the Data Exchange or seamlessly integrate third party data into their warehouse to enrich their SAP data.

6. **Usage-Based Pricing**

Unlike the high fixed licensing costs associated with traditional data warehousing solutions like SAP BW and HANA, Snowflake has adopted a usage-based pricing model. This means enterprises pay for the actual compute and storage they use. That means instead of large upfront investments, costs scale with usage. This is particularly beneficial for businesses with fluctuating workloads, as they don't overpay during periods of lower activity.

For budgeting and financial planning, Snowflake's tagging capabilities give clear visibility into costs, allowing for more predictable budget forecasting.

7. **Automation and Ease of Use**

Snowflake has been designed with user-friendliness in mind. Many tasks that require specialist skills in systems like SAP are automated in Snowflake. This not only cuts down on operational costs but also speeds up processes and allows companies to eliminate the risk of a dependency on highly specialised expertise.

Snowflake can address the common complaints of a more rigid architecture and design through a combination of its cloud-native architecture, allowing separation of storage and compute and through compelling features that allow for greater flexibility such as support for non-structured data and automatically scaling based on workloads.

By focusing on a philosophy of openness and integration, Snowflake provides a compelling answer to many of the integration pain points experienced by SAP customers. The platform's data sharing and broad ecosystem support ensure that businesses can maximise the value from their SAP data while benefiting from the flexibility and innovation that comes with a wide array of third-party tools.

With a usage-based pricing model and features that support robust financial governance Snowflake customers have a transparent and auditable cost structure that provides greater financial agility. The cloud services layer and ease of us further mitigate the costs associated with niche expertise required to build and maintain solutions.

Comparative Overview

It is worth noting that SAP's own solution, SAP Datasphere, has emerged to address many of the challenges SAP customers face.

While the Datasphere looks promising and looks to offer many of the features that are present in Snowflake today, it is currently less mature when compared to Snowflake. As the technology landscape evolves, it's essential for decision-makers to weigh the advantages and challenges of each platform to make an informed choice.

The following comparison table is based on our own experience and expertise.

	SAP BW/4HANA	**SAP Datasphere**	**Snowflake**
Deployment	Self-hosted	Cloud-native	Cloud-native
Pricing	Upfront licence	Usage-based	Usage-based
Ease of use	More complex	Easier to use	Easier to use
Architecture	Shared-nothing	Shared-nothing	Hybrid shared storage & shared-nothing

Primary Storage	In-memory	In-memory	Object storage
Secondary Storage	Disk	Disk	Disk
Features	More mature platform	Less mature platform	More mature platform
ETL Tool Support	Native tools + some third-party support	Native tools + some third-party support	More third-party support

Chapter 3: Discovering SAP Analytics Use Cases

Seamless integration between SAP and Snowflake requires strategic planning. With the right foundation, companies can harness the power of Snowflake to leverage their SAP data in more advanced and efficient ways. This chapter explores the initial steps that need to be taken to ensure that all analytics use cases are identified and a plan is created to deliver these in a Snowflake environment.

Understanding Analytics Use Cases

Start by assessing the analytics use cases supported by the existing SAP solution. Additionally, recognize new analytics use cases vital to the business. This involves:

- Grasping the business processes underpinned by different analytics use cases.
- Identifying the current reports and dashboards, as well as the data contained within.
- Understanding the decisions driven by this data.

By collaborating with key business stakeholders, you can gauge the importance and complexity of each use case. This knowledge will guide the prioritisation of use cases in the Snowflake data platform.

Source System Inventory

It's essential to understand the data sources and their interactions:

- Start with the current systems landscape, understanding key components of source system integrations.
- Identify data load frequencies from each source system, the average data volume, and any relevant processes.
- There is often important and relevant data existing outside of SAP, with other systems or even individual files. Catalogue all data sources and types for integration into Snowflake.

Data Lineage & Data Model

Once the sources of data are known and understood it's important to understand how the data from those source systems ultimately ends up in the final reporting layer:

- Analyse the initial extractors or tables that are used in the initial landing layer of the existing SAP analytics solution.
- Map out how these initial tables travel through the different layers of the data warehouse.
- Document and estimate the complexity of the data transformations and jobs to move data between each layer of the data warehouse.
- Create the high-level data model mapping out the transactions and business processes (facts) and the core business entities involved in these transactions (dimensions).

Planning

Now armed with an understanding of how the analytics use cases are used by the business and how these are technically underpinned the overall implementation can be planned:

- Using the prioritisation matrix determined with the business and validated with the data lineage plan out the order of delivery and estimations for the project
- Determine the resource mix that is needed to deliver the project based on key milestones and times required with dependencies between deliverables factored in
- Identify any key pieces of SAP analytics functionality which is currently used which may not be available out-of-the-box in Snowflake and any technical deliverables that might be required
- Create a decommissioning plan to determine when any existing technology can be decommissioned and the savings that this would mean which can be factored into the benefits realised for the project.

Summary

In this chapter, we delved into the intricacies of preparing for a successful integration of SAP data into the Snowflake platform. Key to this process is a deep understanding of your current analytics use cases, the sources of your data, and the quality of that data. We highlighted best practices such as data profiling, cleansing, and cataloguing, and underscored the importance of mapping out a comprehensive implementation plan.

This plan encompasses aspects from scheduling and task allocation to data transformation, extraction, and validation. By adhering to these outlined best practices, organisations can ensure a smooth and efficient transition of their SAP data onto Snowflake. As we move forward, the next chapter will guide you through the practical steps of executing this integration, focusing on data extraction, transformation, and loading, complemented by strategies for thorough validation and testing.

Chapter 4: SAP Data Extraction

SAP Data Structures and Formats

SAP is a complex system developed over decades, with a diverse range of data types and formats. Grasping these structures and formats lays the groundwork for a seamless integration between SAP and Snowflake.

Originally launched in 1972, SAP's foundational data structures, present in modules such as Finance, Material Management, and Sales & Distribution, remain consistent. During its early development, space optimization was crucial due to database constraints.

As a result, many table names in SAP, stemming from German abbreviations, are restricted to a mere 4 characters, with field names limited to 6. This gives rise to seemingly cryptic table names like LFA1, VBAP, and MARA. Considering a standard SAP system houses hundreds of thousands of such tables, it's no wonder constructing intuitive reports on top of them is a formidable task!

In their quest to simplify analytics, SAP rolled out SAP BW, introducing 'business content extractors'. These extractors, known interchangeably as S-API extractors, SAP data sources, or BW extractors, bridge the gap between the raw SAP tables and data consumers. For instance, leveraging the 'Sales Orders' extractor saves you from wrestling with complex table relationships.

To further aid data analytics, SAP unveiled 'CDS Views' (Core Data Services). Unlike S-API extractors that are rooted in SAP's ABAP programming language, CDS Views utilise SQL with SAP-specific annotations. Not only do they simplify data access, but they also cater to an array of functions including SAP application development, embedded analytics, and data extraction processes.

Common SAP Data Management Obstacles

Navigating the complex data structures and unique formats of SAP presents distinct challenges. Here are some key hurdles faced when managing SAP data:

1. **Data Quality**: SAP's intricate nature can make data upkeep challenging, impacting the reliability of analytics and reports.
2. **Data Integration**: The proprietary nature of SAP's data formats, coupled with strict licensing terms, complicates the integration with external systems.
3. **Data Transformation**: SAP data often requires significant adaptation to suit the requirements of other systems.
4. **Data Security**: SAP databases frequently contain confidential information, necessitating rigorous security measures and adherence to data governance standards.

5. **Expertise**: The nuanced landscape of SAP makes finding skilled professionals a daunting and often expensive task.

SAP Data Extraction

Seamless integration of data from SAP systems into Snowflake calls for a well-planned data extraction process. Designed to reduce latency and preserve data integrity, effective extraction involves understanding and utilising various methods based on an organisation's specific needs.

Key Extraction Approaches:

There's a variety of extraction interfaces, each bearing unique strengths and challenges related to SAP system workload, latency, and incurred costs. Complex IT landscapes often necessitate the amalgamation of two or more techniques for optimal outcomes. These approaches can be largely categorised as:

- **SAP-Supported Frameworks and Technologies**:
    - Operational Data Provisioning (ODP)
    - Access via the SAP Application layer
    - OData interface
    - ODBC/JDBC interface
    - SAP Replication Server (either trigger or database log-based)
- **Third-party Proprietary Solutions**:
    - Database trigger/log replicators
    - Application trigger/log replicators (The specifics of these solutions, while not extensively public, revolve around the assimilation of third-party proprietary software within the SAP ecosystem).

Delving into Extraction Patterns:

The act of pulling data from SAP centres around selecting extraction patterns. For SAP systems, three primary patterns exist: SAP ECC, SAP S4/HANA, and SAP BW.

- **Classic SAP ECC Pattern**: Tailored for organisations reliant on SAP ECC, this pattern leverages SAP's standard extractors to mirror SAP ECC's foundational tables.
- **SAP S4/HANA Pattern**: Ideal for organisations utilising SAP S4/HANA predominantly, it replicates the base tables of SAP S4/HANA using SAP's orthodox extractors.
- **SAP BW Pattern**: Crafted for organisations where SAP BW reigns supreme, this pattern extracts data leveraging tools like SAP LT Replication Server (SLT), Operational Data Provisioning/Operational Delta Queue (ODP/ODQ), SAP Data Services, or SAP BW OpenHub Service.

Selecting the Right Extraction Pattern:

Selecting the right pattern is key for successful data integration. We recommend the following approach when deciding on an extraction pattern.

1. **Data Source Insight**: Gain a deep understanding of your organisation's data sources and their utilisation. This insight aids in choosing the ideal extraction pattern.
2. **Migration Objectives Alignment**: Align the extraction pattern with migration goals. For a migration from SAP ECC to Snowflake, the Classic SAP ECC extraction pattern might be the most suitable choice.
3. **Performance & Scalability**: Gauge the scalability and performance of potential patterns. Some might be better suited for granular datasets.
4. **Security & Compliance Considerations**: In a world rife with data regulations, ensuring that the extraction aligns with standards like SOC 2, PCI DSS, HIPAA, and GDPR is non-negotiable.

A Glimpse into Real-world Extraction Successes:

At Snap we have real-world experiences across industries that highlight the significance of making an informed choice and there is no one size fits all approach, some examples are:

1. A manufacturer with SAP ECC as their cornerstone opted for the Classic SAP ECC pattern, translating to a seamless Snowflake migration.

2. A retailer, with SAP S4/HANA at its core, harnessed the SAP S4/HANA pattern, ensuring a swift and effective Snowflake migration.

3. For a healthcare entity grounded in SAP BW, the SAP BW pattern was the key to ensuring minimal operational disruptions during their migration to Snowflake.

The success of migrating from SAP to Snowflake heavily hinges on the chosen interface and extraction pattern. By deeply understanding data sources, assessing performance metrics, and ensuring regulatory compliance, organisations can set themselves up for a successful implementation.

Chapter 5: Loading and Transforming SAP Data in Snowflake

When building analytics solutions, the integration between systems is paramount to achieving reliable and actionable insights. This chapter delves into the intricacies of importing SAP data into Snowflake and preparing it for analytical queries.

Key Considerations for Data Loading:

- **Choosing the Optimal Loading Method:**
  Snowflake is versatile and supports multiple data loading methods. Whether it's bulk loading for large datasets, continuous loading for ongoing data feeds, or real-time streaming for instantaneous updates, the key is to identify what aligns best with your organisation's data requirements. Look into:

    - **Volume:** How much data are you ingesting?
    - **Complexity:** Does your data come with intricate hierarchies or relationships?
    - **Latency:** How fresh does your data need to be?

- **Configuring the Loading Procedure:**
  From establishing the link between the SAP data source and Snowflake to defining the loading rules, this stage is all about optimization. Improve loading performance your

data loading by:

  - Utilising parallel processing to divide and rule.
  - Optimising network bandwidth for faster transfers.
  - Making informed decisions on specific loading options for your dataset.

- **Validating Your Data Load:**
  As the adage goes, garbage in, garbage out. A core component of this process is data validation. Integrate checks and balances during the loading stage. By nipping data quality issues in the bud even before the migration begins, you're ensuring a smoother analytical process down the line.

Data Transformation

SAP tables are highly normalised and interconnected. To extract value from it, it's essential to transform this data in a way that's ready for analytical workloads.

Navigating Complex SAP Data Structures:

- **Challenge:**
  Business objects, such as sales orders, are often made up of numerous tables. Deciphering and transforming these underlying tables, especially with their intricate naming patterns, into an analytics-ready data model can be challenging.

- **Solution:**
  Familiarise yourself with the SAP data model specific to the modules in use. Utilise the SAP data dictionary and tools like SE11 or SE93. Where they exist then try to use the SAP delivered extractors which already provide data in an analytics ready format and take care of many of the complex joins between tables that are required.

  If you're looking to fast-track this process, consider seeking assistance from specialists or leveraging accelerators from Snap Analytics.

Customizations & Z-tables:

- **Challenge:** Many SAP implementations have custom tables (often starting with the letter "Z") and fields added to standard tables. These customizations can vary between different SAP installations.

- **Solution:** Work closely with SAP developers and system administrators to understand the purpose and structure of custom objects. Maintain documentation of custom fields and tables for future reference.

The act of loading and transforming data is more than just a technical procedure; it's an

interplay of understanding the source system, the target platform, and the underlying business processes. This chapter emphasised the need for a deep knowledge of these elements to successfully build a transformed data model that is ready for analytics.

As emphasised in the last chapter on choosing the right extraction method, making informed choices regarding data loading and transformation stands central to the success of your SAP to Snowflake implementation.

Chapter 6: Optimising Workloads on Snowflake

In this chapter, we'll provide guidance for optimising workloads in Snowflake. The way we will approach this is to first consider how to set up our tables and views for efficient performance and then how do we optimally assign the compute resources to execute those queries.

Setting up tables for efficient query performance

- **Leveraging Micro-Partitions in Snowflake:** Micro-partitions are small, self-contained units of data that are stored together on the same physical storage node. This makes it possible for Snowflake to quickly access and process the data in micro-partitions. They play a pivotal role in ensuring efficient query performance by narrowing down the data segments to be scanned.

  A cost-effective approach to maximising the benefits of micro-partitioning, without incurring additional costs associated with clustering keys, is to naturally cluster your data upon ingestion. For instance, with sales order data, an inherent sequence might be the transaction date. Thus, when inserting this data into Snowflake tables, if we order by the transaction date, we ensure that Snowflake partitions the data in alignment with its inherent order, optimising query performance.

- **Defining Clustering Keys in Snowflake:** Snowflake can automatically re-organise data at its storage layer based on designated clustering keys. While this introduces an added cost, its benefits can be seen when querying very large tables. Additionally, clustering keys are not just confined to tables; they can also be applied to materialised views. Clustering materialised views becomes advantageous when faced with varied data access patterns necessitating distinct clustering keys. When employing clustering keys on materialised views, it's advisable to remove the clustering keys on the underlying base table.

- **Materialised views:** A materialised view is a pre-computed data. Because the data is pre-computed, querying a materialised view is faster than executing a query against the base table of the view. This performance difference can be significant when a query is run frequently or is sufficiently complex. As a result, materialised views can speed up expensive aggregation, projection, and selection operations, especially

those that run frequently and that run on large data sets. It's recommended to use materialised views when the results are a small number of records relative to the base table and the base table does not change frequently.

- **Search Optimization Service:** The search optimization service can significantly improve the performance of certain types of analytical queries. It is particularly useful when performing lookups on VARIANT or GEO data types or point look-ups where the predicate is not a clustering key.

Optimally configuring Virtual Warehouses

- **When to use Clustered Virtual Warehouses:** Clustered Warehouses are designed for high-concurrency workloads. They allow you to automatically scale out by adding more clusters to handle multiple simultaneous queries. If you expect many users or processes to query your data simultaneously, clustered warehouses distribute the load.

- **When to select the right clustering mode:** Clustered Virtual Warehouses can be set to operate in Standard or Economy Mode.

    - **Standard Mode**: Prioritises performance. It resumes and suspends faster but incurs a bit more cost.
    - **Economy Mode**: Prioritises cost savings. Warehouses might take a bit longer to resume, but they are more cost-effective for infrequent workloads.

- **Query Acceleration Service:** The query acceleration service can accelerate parts of the warehouse workload by offloading portions of the query processing work to shared compute resources that are provided by the service.

Examples of the types of workloads that might benefit from the query acceleration service include:

- Ad hoc analytics.
- Workloads with unpredictable data volume per query.
- Queries with large scans and selective filters.

The QUERY_ACCELERATION_ELIGIBLE view can help you identify queries and warehouses that might benefit from the query acceleration service.

Snowflake removes a lot of the traditional performance tuning that earlier generations of data warehouses required but there are still tweaks you can make that have a significant impact on performance and cost optimization.

