

# SAP® HANA Advanced Modeling

Dominique Alfermann  
Stefan Hartmann  
Benedikt Engel

Dominique Alfermann, Stefan Hartmann, Benedikt Engel  
**SAP® HANA Advanced Modeling**

**ISBN:** 978-3-945170-96-0

**Editor:** Alice Adams

**Proofreading:** Tracey Duffy

**Cover Design:** Philip Esch, Martin Munzel

**Cover Photo:** fotolia #50865078 (c) carloscastilla

**Interior Design:** Johann-Christian Hanke

All rights reserved.

1st Edition 2015, Gleichen

© 2015 by Espresso Tutorials GmbH

**URL:** [www.espresso-tutorials.com](http://www.espresso-tutorials.com)

All rights reserved. Neither this publication nor any part of it may be copied or reproduced in any form or by any means or translated into another language without the prior consent of Espresso Tutorials GmbH, Zum Gelenberg 11, 37130 Gleichen, Germany.

Espresso Tutorials makes no warranties or representations with respects to the content hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Espresso Tutorials assumes no responsibility for any errors that may appear in this publication.

**Feedback**

We greatly appreciate any kind of feedback you have concerning this book. Please mail us at [info@espresso-tutorials.com](mailto:info@espresso-tutorials.com).

# Table of Contents

<b>Preface</b>	<b>5</b>
<b>1 Introduction</b>	<b>9</b>
1.1 Intention	9
1.2 Objective	10
1.3 In and out of scope	10
1.4 Content	11
1.5 Data model definition and terms	13
<b>2 Case study and hardware setup</b>	<b>15</b>
2.1 The demand for SAP HANA native solutions	15
2.2 SAP HANA native project introduction	17
2.3 System landscape and configuration	25
2.4 Test approach	26
2.5 Summary and conclusion	30
<b>3 Create the data model</b>	<b>31</b>
3.1 Persistency versus virtualization	31
3.2 Modeling approaches	41
3.3 Engine utilization	51
3.4 PlanViz	55
3.5 Test scenarios	63
<b>4 High-performing information views</b>	<b>85</b>
4.1 Attribute view	86
4.2 Analytic view	96
4.3 Calculation views	130

<b>5</b>	<b>Advanced modeling techniques</b>	<b>189</b>
5.1	Development perspective	189
5.2	Core Data Services	190
5.3	Challenges in SAP HANA modeling	195
5.4	Development methods	221
<b>6</b>	<b>Best practices and recommendations</b>	<b>231</b>
6.1	SAP view strategy	231
6.2	SAP HANA table design	232
6.3	Joins	233
6.4	Calculations	238
6.5	Currency and unit conversion	241
6.6	Filters	244
6.7	General SAP HANA design aspects	247
6.8	Summary	251
<b>7</b>	<b>Operational reporting on SAP HANA</b>	<b>253</b>
7.1	Operational versus strategic reporting	253
7.2	Specifics of operational reporting	255
7.3	Operational reporting scenario	264
7.4	Operational reporting optimization	267
7.5	SAP HANA Live	270
7.6	Summary	278
<b>8</b>	<b>Conclusions and outlook</b>	<b>281</b>
8.1	Conclusion	281
8.2	Outlook	283
<b>A</b>	<b>About the Authors</b>	<b>290</b>
<b>B</b>	<b>Index</b>	<b>293</b>
<b>C</b>	<b>Disclaimer</b>	<b>296</b>

# Preface

This book introduces you to advanced modeling techniques based on actual project experience as well as field tests. Our aim is to provide clear modeling recommendations for setting up data models, optimizing performance, using the SAP HANA modeler and development perspective, as well as efficiently solving complex requirements in SAP HANA.

The target audience of this book is SAP HANA developers and architects acquainted with the methods and options for implementing logic in SAP HANA. The reader should possess a solid understanding of general SAP HANA objects.

In our world, technologies evolve at an extraordinary pace and new market opportunities develop just as quickly. Companies face the challenge of adapting to unfamiliar technologies and markets with different rules and key players. While businesses adapt, IT departments face similar struggles with constantly changing business requirements and new technologies. In these hectic times, it is essential to have a good reporting foundation available to keep track of existing and also new business.

SAP is one of the big players in business software and is currently developing new products and promoting new product suites at a speed previously unknown to most customers. A central element to all of these products is SAP HANA. Most IT departments have seen SAP HANA in sales presentations, or even implemented the software in a proof of concept (PoC). However, few companies truly consider SAP HANA an alternative to existing databases or data warehouses.

In this book, we go one step further and take a closer look at how to build reporting views in an SAP HANA native environment. We not only take a close look at SAP HANA design elements and patterns, but also evaluate these patterns and provide recommendations and best practices for modeling in SAP HANA. The evaluation is based on critical factors such as performance and storage consumption, as well as on ease of use. In the course of the chapters, we examine possible implementation elements or variants in the different types of information views and their impact on the evaluation factors. The evaluation is performed using test scenarios and is based on our practical experience and results in recommendations and best practices. In order to complete the picture, we demonstrate how to solve complex problems and calculations in SAP

HANA. Finally, the book discusses new challenges in integrating operational reporting into the operational system while avoiding redundancy. We hope you enjoy reading this book and collecting insights into SAP HANA native.

Dominique & Stefan & Ben

## Personal dedication

Writing a book like this is also the result of encouraging support and contributions from our friends, families, and colleagues. Their numerous ideas, hints, and discussions helped us to greatly enrich each chapter. We are truly grateful to them. We especially want to mention our employer Capgemini Deutschland GmbH, including Kai-Oliver Schaefer and Thomas Schroeder, who inspired us to go ahead with this book project. Last but not least, we want to say thank you to the Espresso Tutorials team, especially to Alice Adams, who patiently advised us in formalizing and completing the book.

---

We have added a few icons to highlight important information. These include:

### Tips



Tips highlight information concerning more details about the subject being described and/or additional background information.

## Examples



Examples help illustrate a topic better by relating it to real world scenarios.

## Attention



Attention notices draw attention to information that you should be aware of when you go through the examples from this book on your own.

## ⇒ Link codes



For long hyperlinks we include a consecutive number ⇒1 (link code) with the link. Visit <http://4110.espresso-tutorials.com> to find a dynamic list of links that you can click on to easily access information.

Finally, a note concerning the copyright: all screenshots printed in this book are the copyright of SAP SE. All rights are reserved by SAP SE. Copyright pertains to all SAP images in this publication. For simplification, we will not mention this specifically underneath every screenshot.





# 1 Introduction

**In this chapter, we clarify and emphasize the reasons for writing this book. Furthermore, we explain the overall scope of the book and provide a brief introduction to its content. We conclude the chapter by defining frequently used terms.**

## 1.1 Intention

One key element to the successful management of a company consists of a solid business intelligence (BI) solution and highly available business applications. The evolution of new technologies constantly challenges IT management to balance the need for innovation with the demand for consistent and accurately working system processes. One of these new technologies enabled by the decreasing cost of main memory is the in-memory database. In-memory databases not only result in performance improvements for reporting, but also impact the holistic BI strategy. Introducing in-memory databases to the IT landscape offers completely new possibilities to IT and end users and requires a rethinking of existing principles and processes. A further aspect to be taken into consideration for the business is that getting the right information at the right time to the right decision makers becomes increasingly important. Today especially, decisions have to be made in near real time and based on a massive amount of data. SAP is positioning its in-memory database product SAP HANA to fulfill the requirement of real-time reporting and enable new business opportunities. This product represents the foundation for many SAP products and the basis for high-performing data processing and reporting.

SAP HANA delivers a wide range of inherent functionalities that enable it to be leveraged for reporting, making it far superior to the average database. The approach of developing reporting based on the HANA database directly is called *SAP HANA native* development. For this purpose, SAP HANA provides a comprehensive set of out-of-the-box solutions for structured and unstructured data, classical database queries, predictions,

and text or fuzzy algorithms. Furthermore, SAP is continuously improving SAP HANA and with each release new features are distributed to the customers. This in turn results in a large and growing number of possible ways to implement a reporting solution based on SAP HANA native. The idea of this book is to provide a cookbook that you can use to build your high-performing SAP HANA native data model in order to lift your solution into the champions' league of existing HANA solutions.

## **1.2 Objective**

This book introduces you to SAP HANA advanced modeling techniques based on actual project experiences as well as field tests. The goal is to provide clear modeling recommendations for setting up data models, optimizing performance, using the SAP HANA modeler and development perspectives, as well as solving complex requirements efficiently in SAP HANA.

## **1.3 In and out of scope**

In contrast to other publications, the aim of this book is not to provide an introduction to SAP HANA and its possibilities, but to supply the reader with clear recommendations on how to improve both modeling in SAP HANA and creating a development environment on SAP HANA. These recommendations are divided into five areas:

1. Virtualization versus persistency (see Chapter 3)
2. Evaluation of HANA design elements by performance and memory consumption (see Chapters 3 and 4)
3. Advanced modeling techniques (see Chapter 5)
4. Best practice recommendations (see Chapter 6)
5. Operational reporting (see Chapter 7)

The first area examines where persistency is still required in a normally read-optimized in-memory environment. The outcome of the evaluation regarding the second area (2) is a list of design options and their impact on the noted evaluation factors. The third area (3) reviews common challenges during implementations and how to solve these challenges in SAP HANA. Recommendation area four (4) results from the second area (2) and field experience, whereas the final area (5) comprises operational reporting as well as experience from projects.

For the purposes of this book, decision tables, analytic privileges, hardware-related performance tuning, and the consumption of views by different client tools (e.g., SAP BusinessObjects) are out of scope. Furthermore, this book does not explain basic concepts of views or tables in SAP HANA, as an understanding of these concepts is a prerequisite for reading this publication.

## **1.4 Content**

To provide you with a consistent storyline, the book merges a case study with a structured evaluation of SAP HANA design elements and options. After the introduction to the case study (which is referenced in all of the chapters), you will be acquainted with the major considerations for starting your development in SAP HANA in the chapter on the technical foundation (see Chapter 3). Furthermore, we discuss several architectural options that can be applied to the case study. We verify these using test cases which identify the SAP HANA design patterns with the best balance between performance, memory consumption, and maintainability. We walk you through advanced modeling techniques for implementing this architecture and for solving further common business problems. Chapter 6 then covers best practices gathered throughout the previous chapters. The final chapter discusses operational reporting in terms of in-memory databases. The book closes with a conclusion and a review of topics for further investigation. Figure 1.1 provides an overview of the structure of this book.

Chapter 1	Introduction
Chapter 2	Case study and hardware setup
Chapter 3	Create the data model
Chapter 4	High-performing information views
Chapter 5	Advanced modeling techniques
Chapter 6	Best practices and recommendations
Chapter 7	Operational reporting on SAP HANA
Chapter 8	Conclusions and outlook

Figure 1.1: Book structure

Let's take a brief look at the focus of each chapter.

**Chapter 2: The SAP case study** presents the reader with a business case and the (hardware) setup used for the implementation of this business case. Business entities and their correlations are highlighted. To prepare the basis for our test cases, this chapter presents the SAP HANA solution, including the size of the data model and consequently the available hardware. It also discusses a common test approach.

**Chapter 3: Create the data model** looks at data persistency versus virtualization and uses these considerations to create physical tables. This chapter also presents first test scenarios to the reader. These test scenarios evaluate the impact of various physical architectural elements on performance and space consumption.

**Chapter 4: High-performing information views** are structured using the different information views provided in SAP HANA Studio. Through various test scenarios, we take a closer look at the different types of in-

formation views to understand the performance impact of the SAP HANA design patterns provided in these information views. We thereby create a set of guidelines and recommendations for various features of SAP HANA Studio, as well as for the general use of different view types.

**Chapter 5: Advanced modeling techniques** provides detailed insights into the aggregation behavior in SAP HANA which is ultimately the basis for solving difficult calculation problems in SAP HANA. The chapter then focuses on the SAP HANA development perspective, including *Core Data Services (CDS)*, and the use of this perspective for data modeling. Furthermore, this chapter examines common problems within daily work along with solutions to these problems. The chapter concludes by looking at and explaining the topics of parallel development and agility aspects.

**Chapter 6: Best practice recommendations** leverages the insights gained from the previous chapters, multiple test scenarios, and real implementation experience to provide best practice recommendations.

**Chapter 7: Operational reporting with SAP HANA** describes the special requirements resulting from an SAP Business Suite on HANA implementation, focusing on SAP HANA Live as well as reporting. The chapter also examines the new possibilities arising from SAP HANA as an operational database.

**Chapter 8: Conclusion & outlook** summarizes the results and learnings from the previous chapters. Within this chapter, we address how the goals of the book have been achieved. This chapter also gives you an overview of some ideas for further topics that may be interesting to follow up on.

## 1.5 Data model definition and terms

In this book you will come across terms that we need to clarify in advance. In general, we will follow the concepts of the conceptual, logical, and physical data model and clarify the meaning of HANA information views. In the world of SAP HANA native, the concepts of the three different types of data modeling are still applicable. The *conceptual data model* describes the highest level of relationships between different entities, which means that this type of model shows only business terms and how they are connected. No key relationships are shown. The *logical data model* describes the data at a finer level of granularity without taking into

account how it is implemented physically with SAP HANA native. The logical data model already depicts tables and their reciprocal relationships. The *physical data model* reflects the way the data model will be implemented in the database. This includes all table structures and their characteristics, such as column names, data types, key fields, and the table relations. In our case, we introduce an additional split at this level: we add the concepts of *virtual* and *persistent*. The persistent data model comprises everything storing data in a physical way, such as a table. In contrast, the virtual data model does not store the data physically but instead supplies it only virtually when queried, e.g., through information views. The third chapter of this book contains a deep dive discussion on the topic of virtualization versus persistency.

*Information views* are the central components of the solution you develop in your SAP HANA environment. There are three different types of information views in SAP HANA:

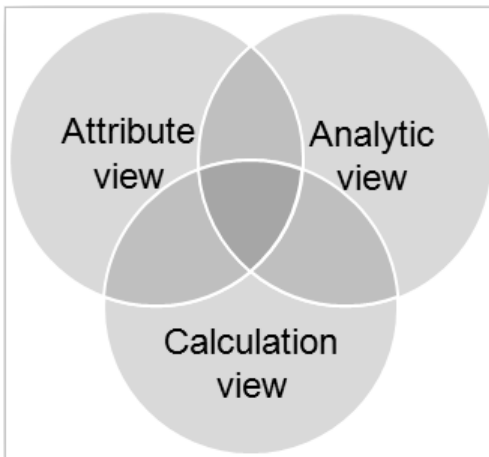


Figure 1.2: Information views

In simple terms, HANA information views are just the metadata definition of how data is transformed. For example, in the view you simply define how a calculation has to be performed; the result of the actual calculation is not yet stored physically within the data. This means that everything is virtualized and all aggregations and calculations are performed on the fly when the view is executed.

## 2 Case study and hardware setup

In this chapter, we use a case study to explain how to implement an SAP HANA native solution. We describe the as-is situation, including known issues with the current business warehouse, develop an entity relationship model, and propose a system landscape. This analysis will serve as the basis for the planned solution. The chapter concludes with a high-level view of our test approach.

### 2.1 The demand for SAP HANA native solutions

Through our daily project work with clients in recent years, we have noticed an increasing demand for real-time reporting and flexible data warehouse solutions. The need for real-time reporting in areas such as logistics, planning processes, market basket analyses, and next best action recommendations emphasizes the numerous use cases. These examples underscore the demand for a wide range of reporting solutions. Given the reporting requirements, clients are likely to invest in these types of technologies. Furthermore, many research facilities regularly analyze reporting needs and user experience. Gartner, Inc. recently published an analysis stating that by using real-time reporting in consumer businesses, for example by sales representatives leveraging data on a daily basis, revenue increased by up to 17%. The corresponding analysis can be found in the article *Fast-Fashion Retailer Boosts Sales With Real-Time Big Data Analysis* by Alicia Fiorletta in Retail TouchPoints.<sup>1</sup>

Based on the fashion retailer Bohme, the article analyzes and highlights the benefits of real-time reporting.<sup>2</sup> According to the article, the retailer achieved a 15% increase in sales shortly after implementing a real-time reporting solution. The company's employees had to deliver an unreasonable amount of work in order to handle warehouse stock and maintain operations. The article states that "tracking and analyzing the sheer

---

<sup>1</sup> <http://www.retailtouchpoints.com/retail-store-ops/2803-fast-fashion-retailer-boosts-sales-with-real-time-big-data-analysis> ⇨1

<sup>2</sup> <http://www.risual.com/retail>

variety and velocity of data became too cumbersome.”<sup>3</sup> Once the company implemented a real-time reporting solution, they achieved a turnaround. Using different visualization types, such as dashboards to report relevant KPIs etc., improved collaboration between the shop floor (production team) and management significantly.

**The development of real-time data for sales representatives had a positive impact on the company’s revenue.**

Best practice emphasizes the importance of real-time reporting for the effective management of companies, as well as for their daily business. Above all, it shows that there is a vast market for real-time data analysis that can be used for different industries and approaches.

Since SAP announced its new SAP HANA technology, these topics have received higher visibility within companies already using SAP, as well as those considering implementing SAP in their IT environment. However, whether or not a company should commit to SAP HANA, particularly an SAP HANA native implementation, is a question that is occupying many decision makers.

SAP Societas Europaea (SE) dedicated itself to this question in a concerted effort with its customers and analyzed their motivations for choosing an SAP HANA native solution. They identified the following key aspects (see Figure 2.1).

**Speed** is one of the main arguments for an SAP HANA native implementation. This was outlined by Mr. Akihiko Nakamura, the Corporate Senior Vice President of the Nomura Research Institute: *“Now and in the future, speed is the key to adapting to an ever-changing business environment. The speed SAP HANA enables is sudden and significant, and has the potential to transform entire business models.”*<sup>4</sup>

In addition, agility, as described by SAP, enables real-time interactions across a company’s value chain.

---

<sup>3</sup> CEO BOHME <http://www.retailtouchpoints.com/retail-store-ops/2803-fast-fashion-retailer-boosts-sales-with-real-time-big-data-analysis> ⇒2

<sup>4</sup> SAP Top 10 Reason <http://www.tyconz.com/more/Top10ReasonsCustomersChooseSAPHANA.pdf> ⇒3



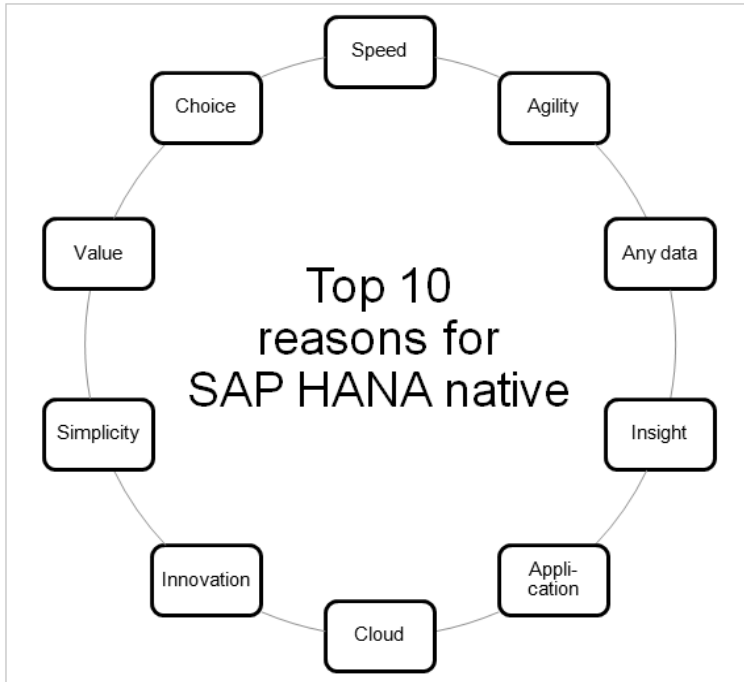


Figure 2.1: Top 10 reasons for SAP HANA native

Another argument for SAP HANA native is the ability to build interfaces for structured and unstructured data. In other words, it enables various heterogeneous data sources to be connected. One of the main benefits of this solution is *insight*, enabling faster response times and consequently, better planning and predictions. Finally, *simplicity* implies the ability to quickly and easily create data flows as long as complex logic is not required. Other aspects such as costs and utilization have to be considered, as well as the return on investment (ROI). SAP clients keep asking questions such as *why should I use SAP HANA native and not some other in-memory solution?*

## 2.2 SAP HANA native project introduction

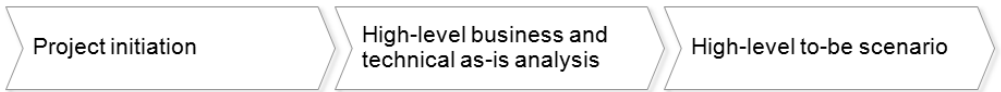
The customer in our case study is a well-known retail company. Their customer analytics department is tasked with focusing on customer behavior. Their analysis looks at a variety of questions, such as the average purchase price per city, repeat customer profiles, and products that sell well.

Our retail company is the market leader in analyzing data and predicting future shopping behavior and trends. They want to sustain a solid market share growth and therefore, the business department intends to implement a real-time reporting solution. In the CIO's opinion, SAP HANA native provides a suitable approach for their end users in the business department.

The IT department is often in conflict with the business department regarding the required reporting processes. The business department intends to analyze the buying patterns of consumers, as brands often invest in marketing campaigns such as discounts and bonus programs in order to incentivize customers to buy their products. However, the most valuable clients are not the one-time purchasers but those buying the brand more frequently. This is actually the most challenging target. For that reason, the business department wants to improve the quality of their analysis in order to increase their performance.

The IT Director decides to set up a new project to implement SAP HANA native in order to provide the business department with the foundation for real-time reporting.

Therefore, the IT Director introduces a simple process grouped into the three clusters noted in Figure 2.2.



*Figure 2.2: Project process*

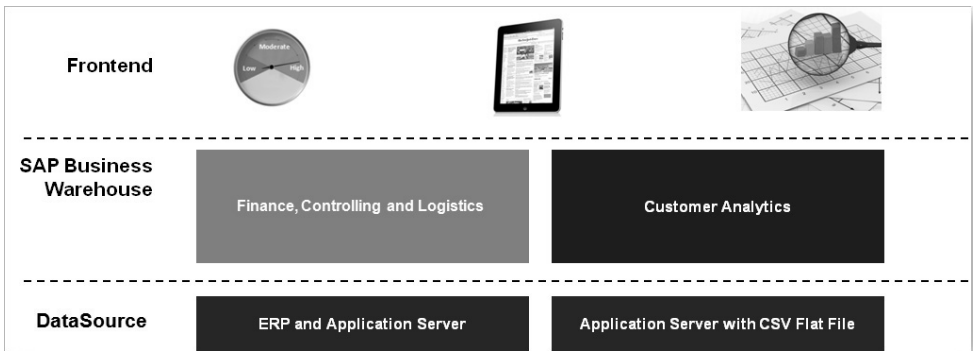
The project initiation represents the project's vision and provides an overall picture of the project. Of course, the IT Director has to convince each stakeholder that he or she is a part of the project and supports the overall vision.

Based on an employee survey and data analysis, the as-is situation is evaluated and the requirements are defined. The final future scenario is described in an entity relationship model (ERM) which is considered the foundation for the SAP HANA native solution.

## 2.2.1 Project initiation

In our case study, the IT Director wants to provide the business department with a best-in-class reporting solution for their daily work. The aim of this project is to replace the existing SAP business warehouse (BW) with the SAP HANA native solution.

Figure 2.3 shows the current system landscape at a high level. We will cover the solution in more detail when we discuss the future scenario.



*Figure 2.3: High-level SAP BW architecture*

The relevant aspect of this project will be the customer analytics data model and the underlying source data. This data model will be reconstructed and adapted to a solution based on SAP HANA native.

The customer analytics will be provided in the SAP HANA Cloud and the new system landscape will be built from scratch. The selected approach should be stable for the next couple of years. To obtain the buy-in of all employees involved, the IT Director demonstrates the strengths of the solution using an SAP HANA native prototype developed within Cloud Services by an IT consulting company. As all stakeholders agree to this solution and the project scope has been defined accordingly, the project can now start.

In the next section, we will take a closer look at the as-is situation. This includes the reporting environment as well as the data itself which serves as the basis for reporting.

## 2.2.2 As-is analysis

The SAP Business Warehouse shown in Figure 2.3 was implemented following the standard Layered Scalable Architecture (LSA) recommended by SAP. During the implementation, many developers worked on enhancements and made adjustments to the data model. As a result, complexity and inconsistency increased dramatically. The Business Warehouse is characterized by various unused objects, complex transformations, and missing documentation.

To understand the weaknesses of the existing solution, the IT Director conducted an employee opinion survey. The results identified six key issues (see Figure 2.4).

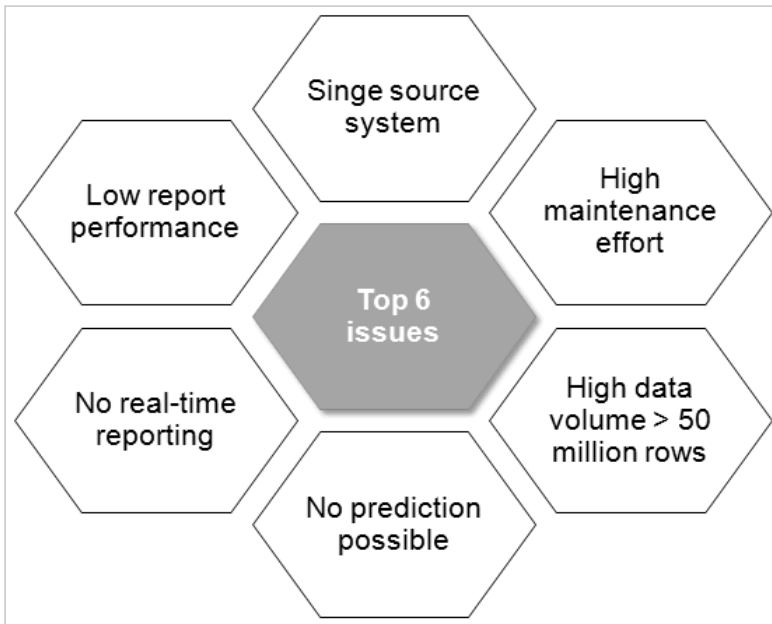


Figure 2.4: Top six issues in the existing IT landscape

Let us take a look at each issue in more detail:

- ▶ **Single source system:** The most common interface on the customer side is a flat file interface. Generally, the data is delivered automatically to a NetWeaver Application Server in a comma separated values (.csv) file format via a software solution.

- ▶ **High maintenance effort:** The complexity of the data model has increased significantly over the years. Therefore, maintenance involves lengthy processing times, time-consuming testing, and requires in-depth knowledge of the data model and the business. Thus, maintenance costs are very high.
- ▶ **High data volume >50 million rows and no real-time reporting:** As a consequence of high data volumes, data loads can only be executed at night (otherwise the system response times are unacceptable). Hence, no real-time reporting is possible and even standard reporting is difficult as response times are mostly above one minute. Furthermore, drilling down to a more detailed level is almost impossible.
- ▶ **Low report performance and user acceptance:** Due to the large volume of data being stored in the system and the complex analyses required, reporting performance is poor. The query execution times generated through the reporting tool were checked and compared to the overall execution time to ensure that the issue was not with the reporting tool. Due to these issues, user acceptance of the system is very low and this is reinforced by actual system usage—the number of reports run daily is very low.
- ▶ **No prediction possible:** Advanced analytics with predictions for future trends is impossible with the current system setup.

### 2.2.3 Implications of the tool selection

Based on these observations, the IT department endorses the proposed SAP HANA native solution as the most suitable solution for the customer analytics team. It is clear that the requested points such as real-time reporting, managing high data volumes, and predictive functionality can be fulfilled by the SAP HANA native solution.

### 2.2.4 Analysis of the source data

The next step is to analyze the different business entities that are currently built in the business warehouse. These entities are used in historical transactions, offers, and current transactions as shown in Figure 2.5.

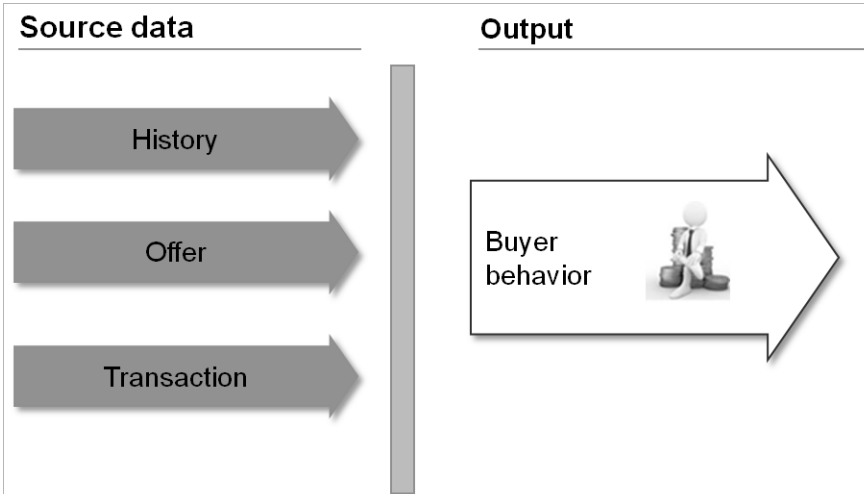


Figure 2.5: Data sources

Each data block reflects one CSV file that is automatically transferred to the system.

The *history data block* consists of a unique customer, a store chain, an offer, and a geographical region. The number of purchases made by a customer is also included. Finally, it contains the date on which the offer was sent out/made to the customer.

The *transaction data block* includes the store chain, as well as an explicit transaction, product category, company, brand number, and the purchase date. The following key figures are also included: product size, product unit, purchase amount, and purchase quantity.

The *offer data block* includes offers, category, brand, and the company number. In addition, the values for the key figures and the quantity of offers are included.

The numerated files contain only anonymized transactional data. To put the data into context, master data is added.

Based on the as-is analyses from the business and technical points of view, the following situation ensues:

- ▶ The IT department initiates the SAP HANA native project.
- ▶ The business is convinced and supports the project's targets.
- ▶ The challenges and expectations are clear.
- ▶ The relevant reporting data is identified.

The IT department proceeds to evaluate the future scenario.

## 2.2.5 Future scenario

Bearing the six main pain points of the existing Business Intelligence (BI) solution in mind, the design of the new backend solution has to ensure easy maintainability, flexibility and simplicity.

As already mentioned, the SAP Business Warehouse will be replaced by the SAP HANA native solution. The significant difference here is that this solution is based in the cloud, which will be explained in detail in the upcoming section (refer to Section 2.3).

Based on the delivered and elaborated data, the IT Director developed an entity relationship model with his team. As a result of the data analysis, the following business entities and their relationships were determined:

- ▶ A customer has an address.
- ▶ A store has an address.
- ▶ A customer makes several purchases.
- ▶ A purchase can include several promotions (e.g., buy one get one free).
- ▶ A store makes several offers.
- ▶ A store sells several products.
- ▶ An offer can contain one or more products.

The ERM model developed serves as a fundamental basis for implementing the data model in SAP HANA Studio (see Figure 2.6).

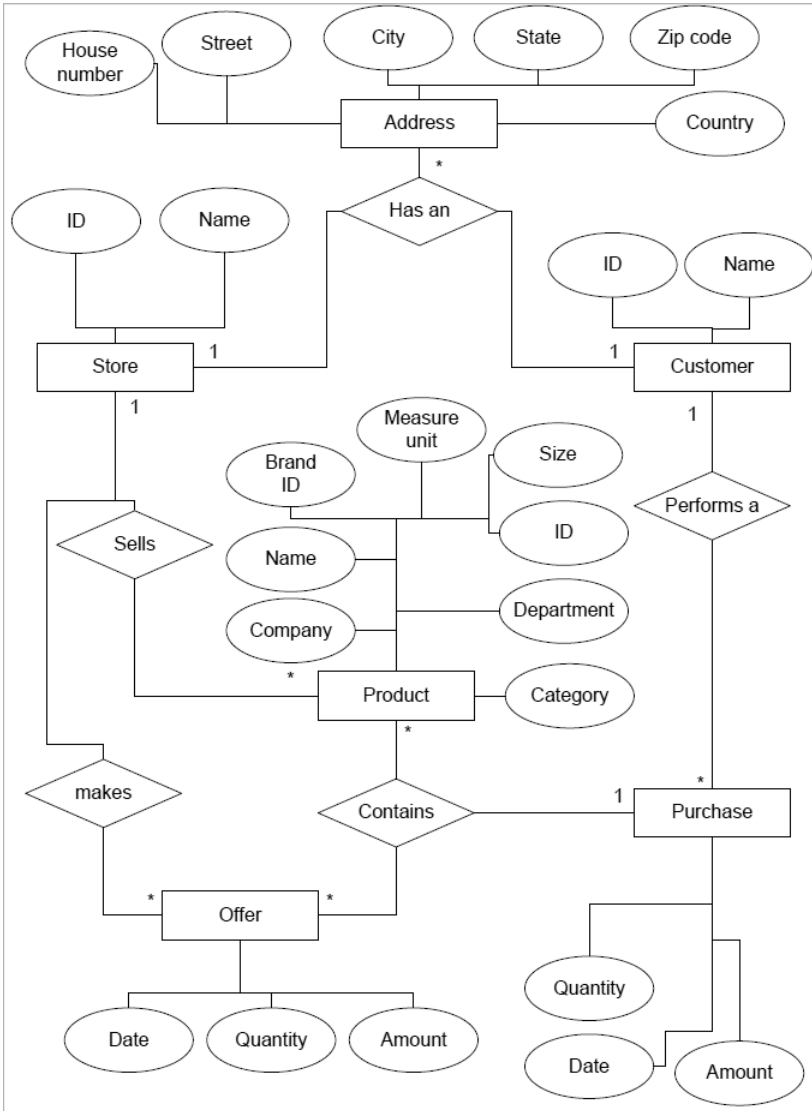


Figure 2.6: Entity relationship model

Using the entity relationship model in Figure 2.6, the IT department fulfills the data requirements identified by the business department. Any enhancements made should keep the model simple and flexible to ensure that the challenges described in the as-is situation do not occur again.



SAP HANA information views are the main method of implementation for realizing the future scenario.

Next, we will walk through the applied technology on a high level to ensure that you have a basis for comparing the tests and implementations with another HANA system.

## 2.3 System landscape and configuration

Since October 2012<sup>5</sup> it is possible to use SAP HANA based on a cloud provided by Amazon Web Services (AWS). SAP is working with Amazon to establish the cloud appliance library (CAL) which makes it possible to run SAP HANA in the cloud. Amazon is one of the biggest cloud computing providers in the world.<sup>6</sup> They are currently building a computer center in Germany. This demonstrates that Amazon is committed to strengthening their position in the German market. Within the scope of these activities, Amazon has to follow the data security guidelines defined by the German government.

In addition to the points mentioned above, one advantage of this solution for the IT department is that costs would be billed hourly. If the business department needs reporting for certain time periods only, the costs can be reduced by shutting down the SAP HANA instance. Due to the advantages mentioned, the IT department decides to use Amazon Web Services (AWS). They select the following configuration as a starting point for providing the business with a suitable solution (see Table 2.1).

Configuration	Key figures
Cores/threads	8/16
Storage volumes	6
Storage (total)	151 GB
Main memory	68.4 GB

Table 2.1: SAP HANA AWS configuration

<sup>5</sup> <http://aws.amazon.com/blogs/aws/sap-hana-now-available-for-production-use/>

<sup>6</sup> Wirtschaftswoche – Der Preis des Terrors; Edition 42, Page 9, 2014

An advantage of this type of solution is that the required hardware can be adjusted at any time. For example, the instance type can be changed to another instance type with more main memory and CPU power with just a few mouse clicks. EBS storage volumes can be added, or the main memory increased easily. The IT department decides that the development should be based on SAP HANA SPS9 and the operation system is a SUSE Linux Enterprise Server 11 SP2.<sup>7</sup>

This configuration was selected because of the expected data volume of approximately 30 GB.

Now that we have explored the future landscape, the next section focuses on the test approach we will use in this book. These tests compare different design patterns in SAP HANA to identify which design elements are most suitable for our purposes and which design decisions have what kind of impact. Design elements can be different information view types, or also simply functionalities within views such as calculated columns. We will evaluate performance impacts and reusability.

## 2.4 Test approach

The test approach presented in this book provides the basis for testing design elements in an SAP HANA native implementation. Since the same goal can be achieved using different designs, the test strategy must ensure that these designs are evaluated in the same way.

To ensure a consistent level of quality and comparability, all test scenarios are structured in the same way and follow a similar process. We will walk you through this process in the following section step by step.

The process is illustrated in Figure 2.7.

Each test will be performed based on the three steps noted in Figure 2.7. First, the scenario and procedure are prepared (1). Next, the tests are executed and the results are documented (2). Last, the test is completed by gathering and comparing the results (3).

---

<sup>7</sup> [https://aws.amazon.com/marketplace/pp/B009KA3CRY/ref=mkt\\_ste\\_sap\\_micro\\_SAPHana1](https://aws.amazon.com/marketplace/pp/B009KA3CRY/ref=mkt_ste_sap_micro_SAPHana1)

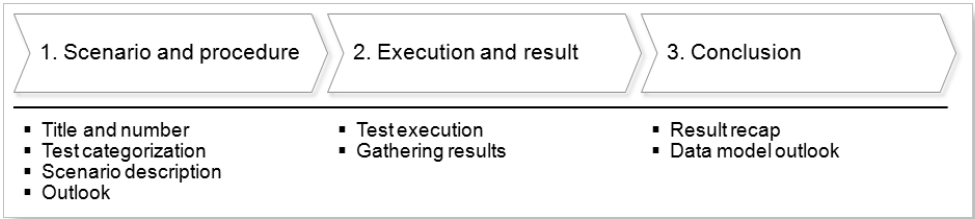



Figure 2.7: Testing process

To prepare the test scenario (1), key metadata including title, numbering, and categorization is provided.

### Test scenario structure



**Structure:**  
Chapter number and test scenario title  
Test scenario number

**Example:**  
4.1.2 Calculated columns in attribute views  
Test scenario 4A.2

In addition, we record the test category, the scenario description, and the expected result in the scenario overview. We also include the architectural elements used to perform the test, e.g., the functionality *calculated columns* in an attribute view (ATV).

Table 2.2 shows a sample test scenario for calculated columns.

<b>Test focus:</b>	Calculated columns
<b>Scenario description:</b>	Analysis of the impact of calculated columns on the performance of attribute views
<b>Expected result:</b>	<ol style="list-style-type: none"> <li>1. An overview of the performance impact of calculated columns by data type</li> <li>2. An overview of the impact of several calculated columns on the attribute view performance, also combining data types in correlation to the findings of (1).</li> </ol>
<b>Architectural elements:</b>	Calculated columns in attribute views

Table 2.2: Test scenario example

Subsequently, we provide a high level overview of the procedure and detail the individual process steps (see Table 2.3).

Step	Testing detail
1	Use the same attribute view several times to create different calculated columns with different formats to ensure that the performance measurement is performed on the same dataset.
2	Combine the calculated columns in one attribute view and measure the outcome.

Table 2.3: Test scenario procedure

The test is executed step by step based on this test scenario. The results are then depicted in a separate table. After the test execution, we summarize the outcome and provide recommendations for modeling your own performance-optimized information view.

Figure 2.8 provides an overview of the test areas in each chapter. The test areas are clustered into data model creation and the performance optimization of information views. The information view area is grouped into attribute, analytical, and calculation views. These areas are then summarized and compared in the last section.

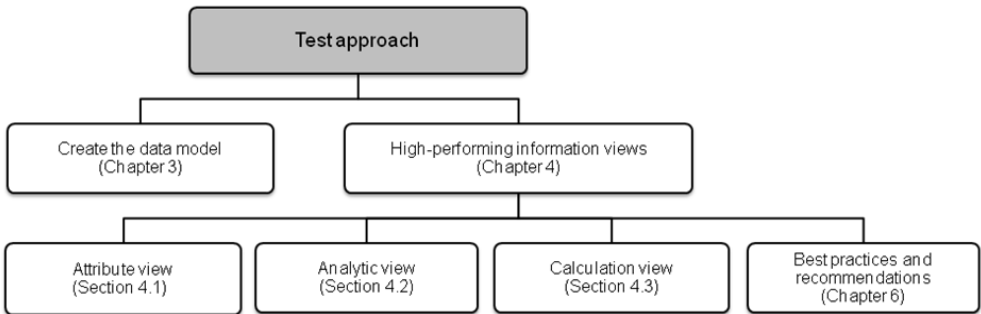


Figure 2.8: Test approach

### 2.4.1 Create the data model

The test area **Create the data model** focuses on recommendations for creating tables in SAP HANA native. Due to the fact that the data model is implemented from scratch, we also have to clarify which data storage

approach should be used for faster reporting. For this reason, we will compare normalization and the star schema. This test area also looks at the join performance of primary keys (PK) and foreign keys (FK), different data types, and unique values with regard to saving space. Accordingly, we will also do a first test on the performance of data loading. In addition, we will compare virtualization and persistency.

## 2.4.2 Optimizing information view performance

Once the data model is successfully implemented, we will continue with tests on information views. In doing so, we will evaluate the implementation of different design elements, such as currency conversion, variables, and input parameters. This will serve as a type of catalog for your future SAP HANA native project as we provide recommendations and design principles for the data modeling based on the test results (see Figure 2.9).

The **attribute view** considers aspects such as the join performance, selections, and filters. Furthermore, different data types and their conversion are evaluated here. The testing around **analytic views** (ANV) covers topics such as the data foundation and investigates the logical joins, including currency conversion, calculation before aggregation, and the different data types. With regard to the data foundation, we analyze the join options.

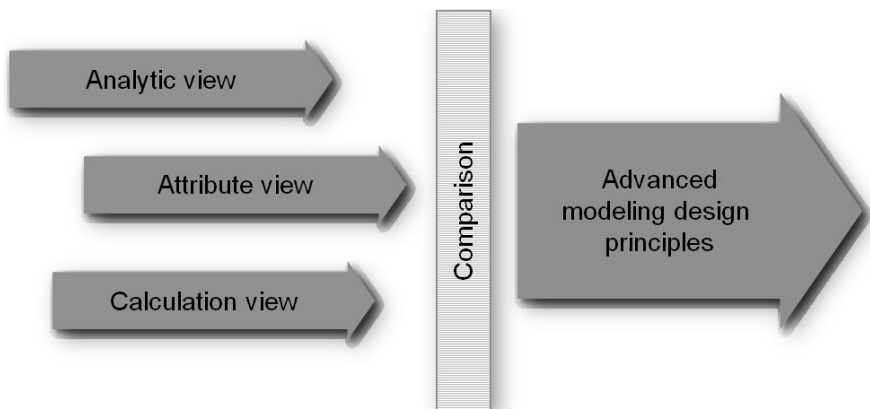


Figure 2.9: View comparison

Next, in the **calculation view** test, we cover different nodes, such as projection and aggregation nodes. In addition, we focus on calculated

columns, metrics, currency and unit conversion, as well as filtering. To conclude the test scenarios, we analyze the different join variants, scripted calculation views, and parallelization in the calculation view design.

The chapter closes with a summary of the results.

## **2.5 Summary and conclusion**

This chapter walked you through the case study and explained the anticipated bottlenecks for the future solution. The ERM developed represents the foundation for the data model implementation within SAP HANA Studio later on.

We also presented the system landscape and explained the test approach.

The next chapter covers the implementation of the data model in detail.

## 3 Create the data model

This chapter covers key concepts for defining an SAP HANA data model. The most frequently asked questions we hear have to do with data virtualization and persistency. In this chapter we elaborate on these topics and take a closer look at the general strengths and weaknesses of both, as well as approaches for creating persistency and the limitations of virtualizations. To create the data model for our case study, we perform the first test scenarios to illustrate the setup of our data model.

### 3.1 Persistency versus virtualization

SAP HANA supports on one hand the physical storage and provision of data in tables, and on the other hand, the preparation and runtime access of data via information views (e.g., analytic view, calculation view). A critical consideration is whether to store data physically (*persistency*) or use the tremendous in-memory power of SAP HANA for runtime calculations (*virtualization*). This question arises whenever you set up a new data model or change an existing one. There are many voices calling for a completely virtual data modeling approach when it comes to reporting. The most common ideas are either to build virtual layers directly on the transactional systems, or create one common replicated data layer as the foundation for what will probably become a virtual enterprise data warehouse. We are convinced that a clever balance between data virtualization and persistency will provide you with the best cost benefit, or cost performance ratio, and we will prove this in the course of this book.

To start the discussion around data persistency and virtualization, we have summarized some of the key arguments in Table 3.1.

	Persistence	Virtualization
<b>Strengths</b>	<ul style="list-style-type: none"> <li>▶ Storage of historical data</li> <li>▶ Provision of historical truth</li> <li>▶ Maintenance of time-dependent data</li> <li>▶ Storage of intermediate results of complex data transformations</li> <li>▶ Reconciliation of data/audit trail</li> </ul>	<ul style="list-style-type: none"> <li>▶ Performance support for (near) real-time queries</li> <li>▶ Sourcing the single version of truth</li> <li>▶ Data consistency</li> <li>▶ Ease of data recovery</li> <li>▶ Sizing requirements</li> </ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"> <li>▶ Limited support of real-time processing</li> <li>▶ Costs of data storage</li> <li>▶ Redundancy of data</li> </ul>	<ul style="list-style-type: none"> <li>▶ Creation and maintenance of complex data transformations</li> <li>▶ Data model management</li> <li>▶ Risk of failure</li> </ul>

Table 3.1: Strengths and weaknesses of virtualization versus persistence

### 3.1.1 Strengths of persistence

This section covers the strengths and benefits of physical data storage per layer.

#### Storage of historical data

One of the fundamental drivers for persisting data comes from the demand for historical data to be available. Transactional systems do not typically provide the entire history of data that may be necessary for reporting or analysis requirements. There is generally no option available to regenerate historical data on the fly. Thus, you must identify a physical location for keeping the requested raw data history available. This can also be a kind of persistent corporate memory as suggested in the SAP *Layered Scalable Architecture LSA++* approach.



## Resources for Layered Scalable Architecture LSA++



Further details on the Layered Scalable Architecture approach in SAP HANA databases (LSA++) are available at

[http://help.sap.com/saphelp\\_nw74/helpdata/en/22/ad6fad96b547c199adb588107e6412/content.htm?frameset=/en/76/0a0f65cca14e369fa8df4b77367b4b/frameset.htm&current\\_toc=/en/ad/6b023b6069d22ee1000000a11402f/plain.htm&node\\_id=12&show\\_children=false](http://help.sap.com/saphelp_nw74/helpdata/en/22/ad6fad96b547c199adb588107e6412/content.htm?frameset=/en/76/0a0f65cca14e369fa8df4b77367b4b/frameset.htm&current_toc=/en/ad/6b023b6069d22ee1000000a11402f/plain.htm&node_id=12&show_children=false) ⇒4

### Provision of historical truth

Building on the previous section, the requirements regarding the provision of historical truth are also possible drivers for storing data physically in their original structures. Essentially, there are three types of requirements for reporting historical data.

1. *Show historical data using (master data) structures valid today.* This approach uses the (physical) representation of the historical raw data and applies the (master data) structures valid at the time of the query (e. g., an organizational hierarchy). In this approach, the structural information is applied to the (physically) stored raw data during runtime. Therefore, no additional physical storage of transactional data is normally required.
2. *Show historical data using (master data) structures valid on the posting date (historical truth).* For this requirement, the historical transactional data is often directly combined and stored together with the structural information valid at a specific point in time (e. g., at the time of posting). Depending on the complexity and type of structural changes, it is also best practice in a HANA-based environment to store the historical structural information physically as part of the transactional data. The main reasons for this approach are maintainability of the data model, reproducibility, and durability of the data. SAP HANA supports this approach with the *history table* concept, which also covers time-dependency of data.

3. *The user decides on a point of time/version of historical data.* This variant is a combination of the second approach with a user-defined parameter. The user chooses a value or key date for which the requested data will be shown.

## **Maintenance of time-dependent data**

When it comes to the time dependency of data, one of SAP HANA's out-of-the-box solutions is history tables. Generating timestamps virtually or on the fly is not something that can be maintained in large analytical scenarios. History tables automatically add timestamps to each data record (VALIDFROM and VALIDTO). Write operations always insert a new version of a record in the table (instead of overwriting existing records). Update and deletion operations also result in new versions with updated timestamps of the existing records. The original records are invalidated in parallel.

## **Storage of intermediate results of complex data transformations**

One of the most crucial points in data modeling is how to best generate and approach the results of complex data transformations. What is a complex transformation? Unfortunately, there isn't an obvious answer to this question (we also don't have an easy answer). However, there are good reasons from numerous SAP HANA implementations for storing the results of complex data transformations physically. SAP HANA is capable of performing every transformation requirement at the point of query execution/data access (at least when you scale your hardware accordingly). We recommend storing (intermediate) results of complex transformations physically, opposing the requirement of a short query response time at one extreme and covering the complexity of necessary transformations, calculations, or enhancements of data at the other.

This applies especially when very short query response times are expected with comprehensive data transformation logics in parallel. However, when it comes to (near) real-time reporting needs, this approach does not fit; the costs over time for storing the intermediate results are inappropriate and too high.

## SAP HANA Enterprise Information Management (EIM)



An interesting approach in this context is provided by SAP HANA Enterprise Information Management (EIM). Available since SPS 9, this option provides valuable features for enhancing, cleaning, and transforming data in order to make it more accurate and useful. Further information and details of current limitations are available at: [http://help.sap.com/hana\\_options\\_eim](http://help.sap.com/hana_options_eim).

### Reconciling the data/audit trail

For quality assurance purposes, data for reporting or analysis demands is regularly compared to the original data (e.g., in a transactional source system). In particular, SAP HANA systems subject to external audits require that data is easily reconcilable and traceable. To achieve reliable, transparent, and sustainable data reconciliation, the physical persistence of data best supports traceability. This is also valid for SAP HANA-based scenarios. Virtual approaches generally do not provide sufficient support for versioning of their definitions because previous data states cannot be restored exactly, and consequently, cannot be audited.

### 3.1.2 Virtualization strengths

As SAP continuously showcases the capabilities and power of SAP HANA, the vision of processing a huge quantity of data on the fly is on everyone's mind. This section highlights how virtualization can empower your data model in this regard.

#### Performance—support of (near) real-time queries

SAP HANA enables high-performance operations on inconceivably huge volumes of data. Its column-based approach best supports read operations on large data volumes. That is why column-based tables are used

for reporting and analysis requirements. The downside of this approach is the reduced speed of insert and update operations. Thus it is best practice, especially for the realization of (near) real-time reporting scenarios, to implement all necessary data transformations and calculations only with SAP HANA's virtual concept of information views. The costs of any physical write or update operation (e.g., to store the result of a complex data transformation) might lead to a failure of the expected query response time. In any case, the demand for (near) real-time response time has always been referenced with the necessary complexity of data transformation. The more complex the transformation logic for preparing the data for the expected outcome, the less likely that (near) real-time reporting will be achievable (even in a completely virtual scenario). A (near) real-time scenario can nevertheless combine a physical real-time-capable data replication (e.g., by using *SAP Landscape Transformation Server (SLT)*, *Sybase Replication Server (SRS)*, or *SAP HANA EIM*) with data transformations and calculations implemented in information views on top of the replicated data layer.

## **Sourcing the single version of truth**

Searching for the single version of truth is an ongoing challenge in today's information landscape. Each and every time data is copied, transferred, or transformed there is a potential risk of failure. Failures can be of a technical nature (e.g., outage of a source system), or can occur on the application level (e.g., faulty transformation rule, missing look-up information). It takes significant effort to identify the root causes of data mismatches and to (re-)establish the desired single version of truth. The fewer physical copies and layers used to prepare the data for its intended use, the fewer sources of error that exist. Virtualization, especially when paired with the in-memory capabilities of SAP HANA, can strongly support achieving a single version of the truth. SAP HANA can combine various data operations within a single information view and can execute multiple transformations in a sequence to provide data in the requested output format. It is no longer necessary to design step-by-step transformations with several physically stored intermediate results in between. Thus, virtualization in in-memory environments leads to fewer layers in the data model and to less physical storage and duplication of data. Using data replication engines such as SLT or EIM is a good complement to virtual concepts for data transformation. The one-to-one replication of source data into a reporting environment builds a solid and (in terms of transaction security) valid foundation for setting up virtual layers on top acting as a central entry point to access the single version of truth.

## Data consistency

Traditional, relational databases provide transaction security according to the *ACID principle* (atomicity, consistency, isolation, and durability). In reporting environments, many sources (e.g., database tables) are combined to display the complete and correct result of a reporting request. Therefore, the data of involved sources is normally physically copied, transferred, and transformed accordingly. For each source, these data operation processes must successfully complete (database commitment for each table) before any database query can be executed. As we have learned, virtualization in in-memory environments helps to reduce the number of (physical) layers in a data model and, therefore, reduces the number of physical database operations. Furthermore, virtualization enables direct access to source tables. In this scenario, the correctness of the data across the entire solution is based on the transactional consistency of the source system. The source system is the foundation for all data transformations (e.g., implemented in information views) to generate the expected result. From a technical perspective, *Smart Data Access (SDA)* is a promising approach for directly accessing source system data with a remote database connection (no physical copy of data in the target environment). The need to ensure overall transactional security is thereby pushed down to the source level.