HA100

SAP HANA Introduction

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 14
Course Duration: 2 Day(s)
Material Number: 50144112

SAP Copyrights and Trademarks

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	Example text
Window title	Example text

Contents

Course Overview

TARGET AUDIENCE
This course is intended for the following audiences:

- Application Consultant
- Project Stakeholder

UNIT 1 Describing SAP HANA

UNIT OBJECTIVES

- Understand the need for a modern digital platform
- Describe how SAP HANA powers a digital platform

Unit 1
Lesson 1

Understanding the Need for a Modern Digital Platform

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the need for a modern digital platform

Today's Digital World

In a changing world we are more connected than ever before, with more valuable data being generated every day. More services are moving online, and this trend is set to grow exponentially. Forward thinking organizations are already taking steps to adapt to the new digital world and grow their businesses.

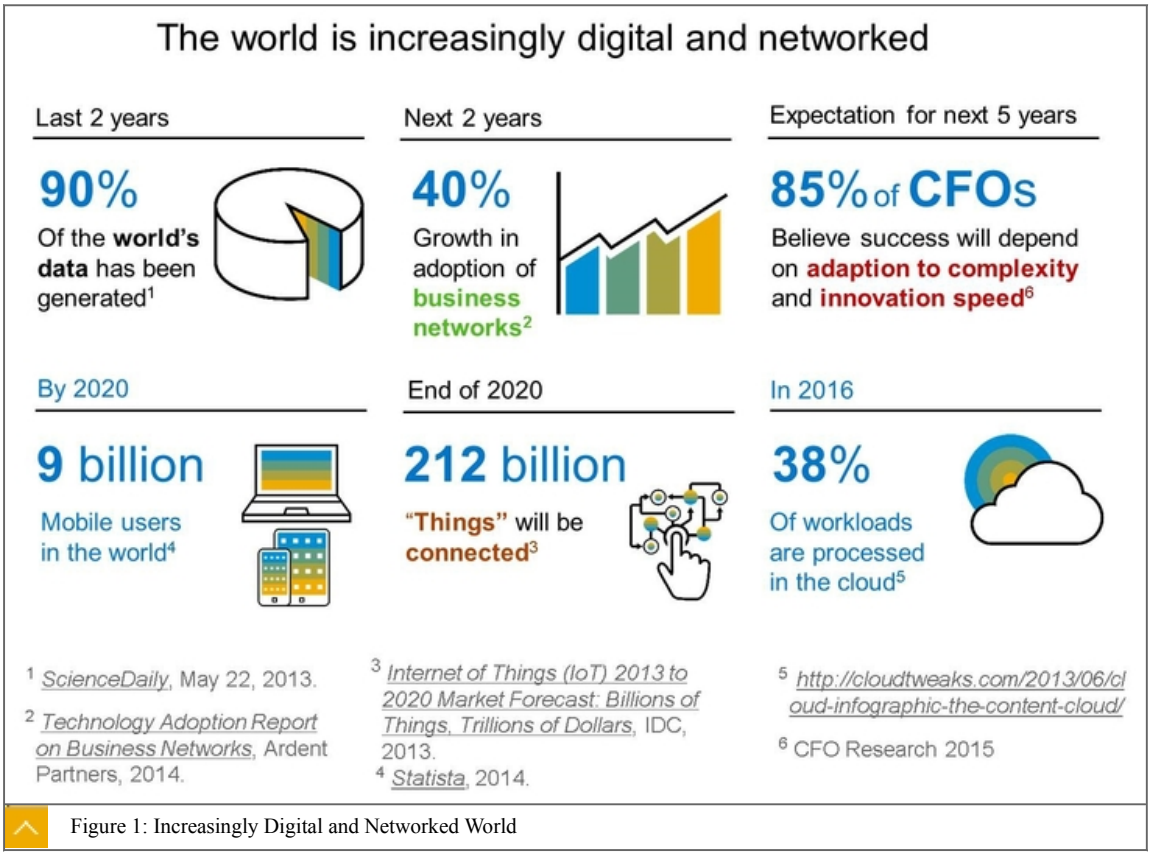


Figure 1: Increasingly Digital and Networked World

The exponential growth of mobile devices, social media, cloud technologies, and the data they generate has transformed the way we live and work. 61% of companies report that the majority of their people use smart devices for everything from e-mail to project management to content creation.

All of this creates unprecedented opportunities for all organizations to grow their businesses. They can do this by exploiting the connectivity of consumers and business partners, tapping into the depth and variety of new types of data, acquiring this data in real time for real-time decision making, and developing innovative new applications quickly.

Increasingly Digital World

Consumerization is driving expectations of what business IT should offer for its users. As business users become familiar with smart consumer applications in their private lives, they also demand real-time, innovative applications. These applications are needed to enable deep insight and provide proactive decision support in their jobs.

However, current business systems cannot cope with the increasing demands of the digital world. We cannot just keep adding more complexity to existing IT landscapes in the hope that we can keep pace with trends. What is needed is a fresh start. It is time to start with a blank canvas and rebuild the business systems from the bottom up. This fresh start must use only the latest technologies aligned to the modern digital world. This is exactly what SAP have done by developing SAP HANA, a brand new platform built for the digital world.

Stifled Growth due to IT Complexity

Typical IT landscapes have developed over time into multiple complex arrangements of purchased, acquired with developed applications, or powered by multiple platforms. These platforms can be based on incompatible hardware from different suppliers. This can mean different operating systems and different databases, and even different development languages. To pull together these different applications, SAP have added extra applications and created interfaces between systems.

Current Landscapes: Problems

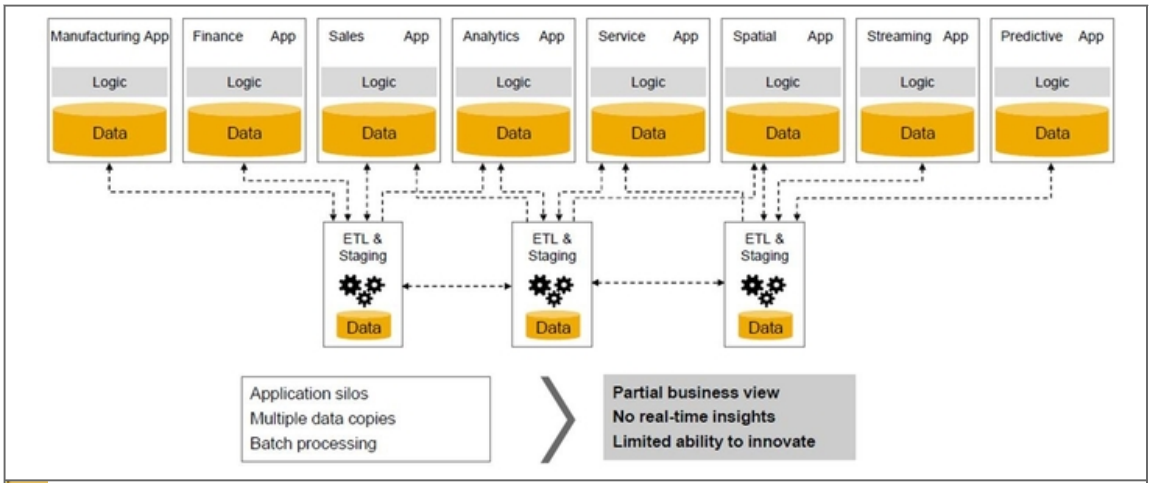


Figure 2: Current Landscapes: Problems

The IT department has been responsible for the integration of these systems. Moving, harmonizing, and cleaning data results in multiple copies of that data. We have placed huge demands on system resources during batch processing. This means that users have to wait for long running processes such as financial close, consolidations, and Materials Requirement Planning (MRP).

Complex landscapes create fragmented business views of data. To obtain a holistic view, users are required to wait until consolidation is complete. Developing new applications in a complex landscape is also difficult. It takes time and is expensive to build and maintain.

There is too much IT complexity in most organizations. Complex landscapes are costly to maintain with multiple skills needed.

Complexity is stifling growth and suppresses agility and innovation, which is critical to survival in today's digital world.

One Platform for all Applications

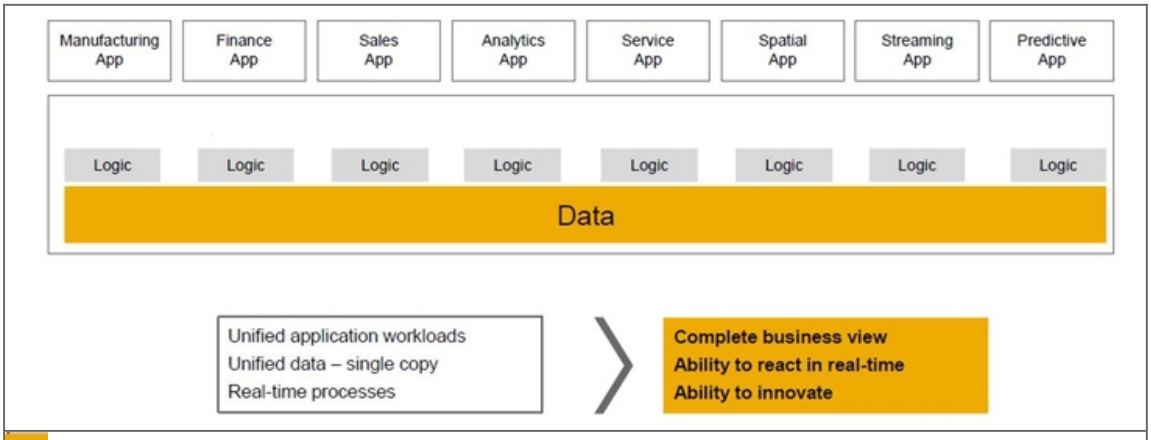


Figure 3: One Platform for all Applications

The answer is to have all applications powered by one high performance, multipurpose platform. This means a common architecture with only one store for all data, regardless of data type. Data is available to all applications in real time. This means that there is no more data movement or management of multiple data stores.

Note:
Later in the course we discuss multitenancy and how this is used to isolate multiple application running on one SAP HANA platform.

Consolidation of OLTP and OLAP

A key objective of SAP HANA is to remove all redundancies. This means that only one copy of data is needed for any type of access.

Traditionally, systems were either optimized for transactions (OLTP) or analysis (OLAP). Transactions were managed in systems where both the hardware architecture and database design, and the data models were built around fast read/write at the record level. Analysis systems took on a different design approach. The hardware, database, and data models were built around batch loading, aggregated storage, and a focus on read-intensive queries.

That is why, historically, OLTP and OLAP were separated. Now SAP HANA takes on the challenge of consolidating transactional and analysis requirements in one platform.

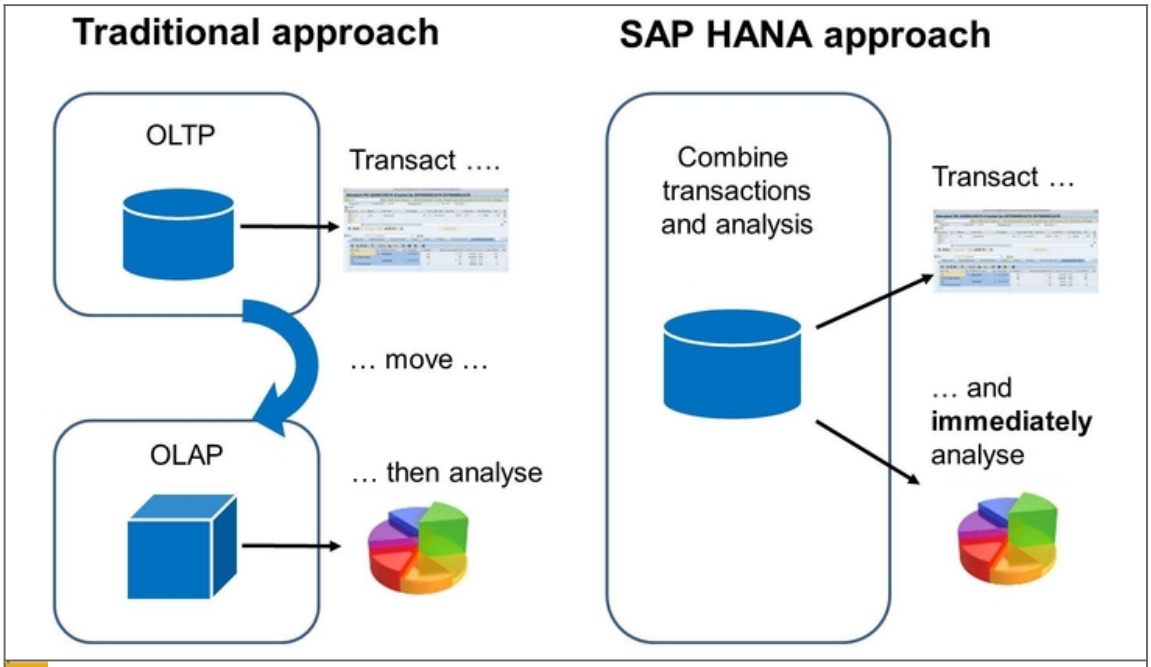


Figure 4: Consolidating OLTP and OLAP

SAP HANA takes on the challenges of combining transactional and analysis processing in one platform. The database, hardware, and data model of SAP HANA are built for combined transactional and analysis purposes. No movement of data is necessary and you always work from the same, single copy of the data for any requirement. This is true for both transactional and analytical requirements. This means that you have live data available to all applications. This reduces the complexity by removing the need to move data using separate software.

Exploiting Technological Advancement

Some of the questions arising from the latest release include “How can one platform handle all applications?” and “Why did we not do this earlier?”

SAP HANA has been developed from scratch to take advantage of the recent trends and advances in hardware. This redevelopment was undertaken to ensure that it is able to handle such an ambitious challenge.

Historically, the high cost of memory meant that only small amounts were available to use. This caused a serious bottleneck in the flow of data from the disk all the way to CPU. It did not matter how fast the processor was if the data cannot reach it quickly.

We now have access to huge amounts of cheap memory. SAP HANA runs on hardware with many terabytes of memory. With so much memory available, you can store the entire database of even the largest organizations completely in memory. This gives you instant access to all data, and eliminates wait times. You can lose the mechanical spinning disk and the latency it brings, and rely on memory to provide all data instantly. Memory is no longer the bottleneck it once was. To address large amounts of memory, you also need 64-bit operating systems. Traditional 32-bit operating systems cannot address the large amounts of memory now available.

In addition to huge memory, processors continue to improve at a phenomenal rate. You now have high-speed multicore processors that can take on complex tasks and process them in parallel. This means that response times for even the most complex analytical tasks, such as predictive analysis, can be carried out in real time. Multiple CPUs that support multiple cores now gives you access to huge processing power. You can therefore consume and process

huge volumes of data in minimal time, which provides business users with instant response applications.

Recent Advances in Technology

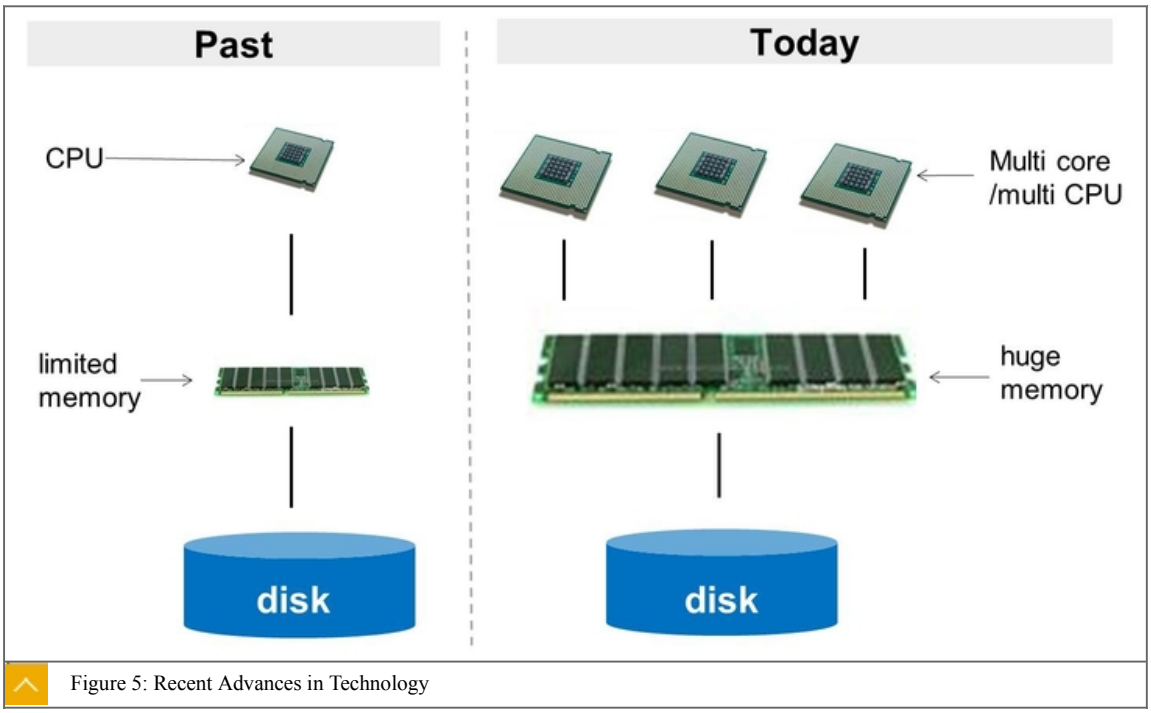


Figure 5: Recent Advances in Technology

Advances in the design of on-board cache mean that data can pass between memory and CPU cores rapidly. In the past, even with large amounts of memory, this created a bottleneck as the CPUs were demanding more data, and the journey from memory to CPU was not optimal. You now have sophisticated on-board CPU cache that keeps the most useful data closest to the CPU, and avoids reading from memory unless absolutely necessary.

With modern blade-server architecture, you can now add more RAM and more CPUs into your landscape easily. This adds more processing power or memory, allowing you to scale up to any size.

It would have been possible for SAP to have kept the same business application software that was written 20 years ago, along with the traditional databases that supported them, and installed all this on the new powerful hardware. This would provide some gains, but traditional databases and applications were designed around old, restricted hardware architecture. This means they would not be able to fully exploit the power of the new hardware, with all the new developments previously mentioned.

Put simply, the business software needed to catch up with advances in hardware technology. Thus, a complete rewrite of the platform (SAP HANA), as well as the applications that run on the platform, was required.

SAP built SAP HANA to fully exploit the latest hardware. SAP collaborated with leading hardware partners who shared the designs of their new CPU architectures. This enabled SAP to develop SAP HANA in such a way that it could extract every last drop of power from the hardware.

LESSON SUMMARY

You should now be able to:

- Understand the need for a modern digital platform

Unit 1
Lesson 2

Describing how SAP HANA Powers a Digital Platform

LESSON OVERVIEW
You want to develop a high level understanding of SAP HANA so that you can assess its potential for transforming existing application and developing new ones.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe how SAP HANA powers a digital platform

Total Database Transfer to Memory

In the past, databases were stored completely on disk. Only the data requested by the applications would be moved to memory, where it then passed to the CPU for processing. Data in memory would be constantly displaced with new data requests, so a lot of swapping was normal. With SAP HANA, you can now store the complete database in memory. This means that no more disk movement is needed and swapping can be eliminated.

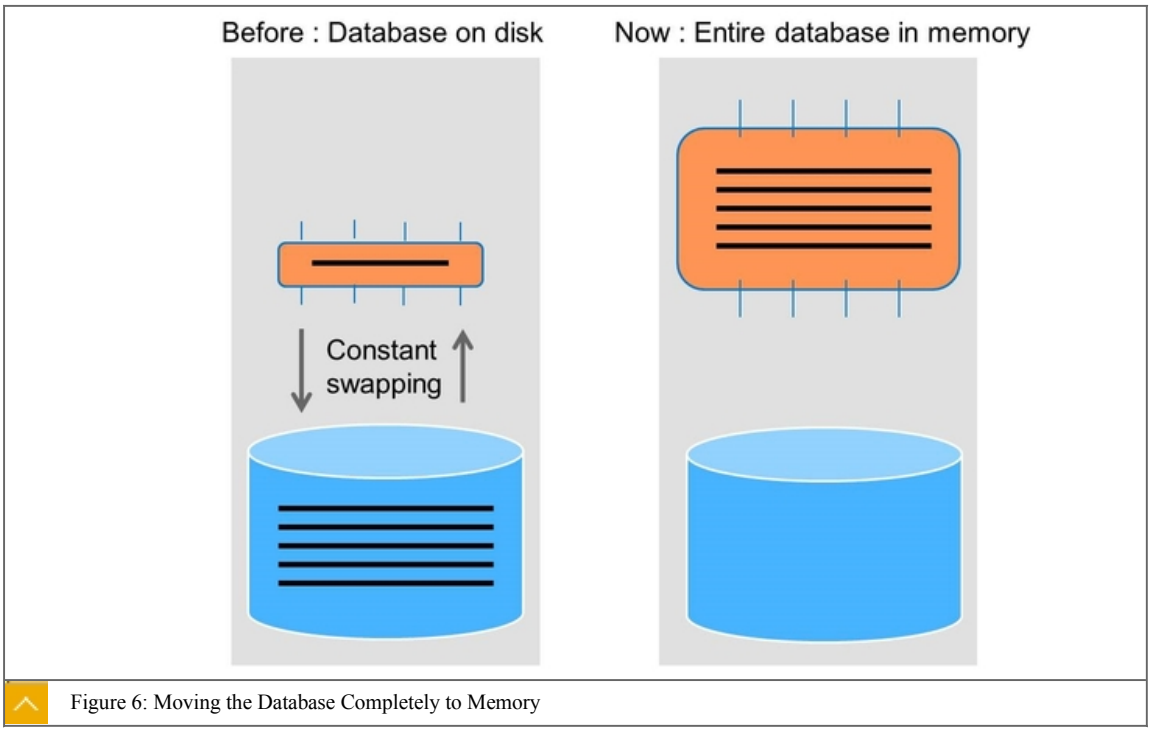


Figure 6: Moving the Database Completely to Memory

You can transfer the entire database to memory due to the following important advances:

- Huge amounts of memory are now available. We have moved from gigabytes of memory to terabytes of memory.
- SAP HANA automatically compresses data. This compression reduces the data footprint of the largest databases down to a fraction of their original size.

Note:
Since 2006, Business Warehouse Accelerator (BWA) also moved large amounts of BW disk-based data to memory to improve read performance. However, BWA could never move the entire BW data to memory, only selected InfoCubes. So, you had to make a choice as to which BW data you wanted to accelerate. SAP HANA "accelerates" all data because all data can fit in memory.

SAP HANA Disk Store

However, this does not mean that disk is no longer needed. SAP HANA includes disk store.

You need disk for the following reasons:

- Data in memory is referred to as "hot", which means it is highly used and needs to be closest to the CPU for optimum read performance. Less used data can be classified as "warm", which means it is stored on disk. SAP HANA will always attempt to store all data in memory. However, most organizations would not want all data in memory as they regard only a part of it to be hot. The warm data can wait on disk until it is needed, at which time it is called into memory. Of course, this means a slight delay in getting data to the CPU when compared to the hot data, but for data that is warm, this is usually acceptable. This means that you can deliberately size memory optimally to fit only the hot data and not worry about trying to fit the entire organization's data in memory.
- Disk is used as a safe backup of memory, in case of power outage. SAP HANA regularly saves the entire contents of memory to disk so that when power is restored, memory can quickly be restored from disk.

Both of these requirements will be covered in more detail later in this course. For now, it is important not to ignore disk. While it is no longer needed for reading data where high performance is required, it does still have its place in SAP HANA.

Simplified Data Models and Applications

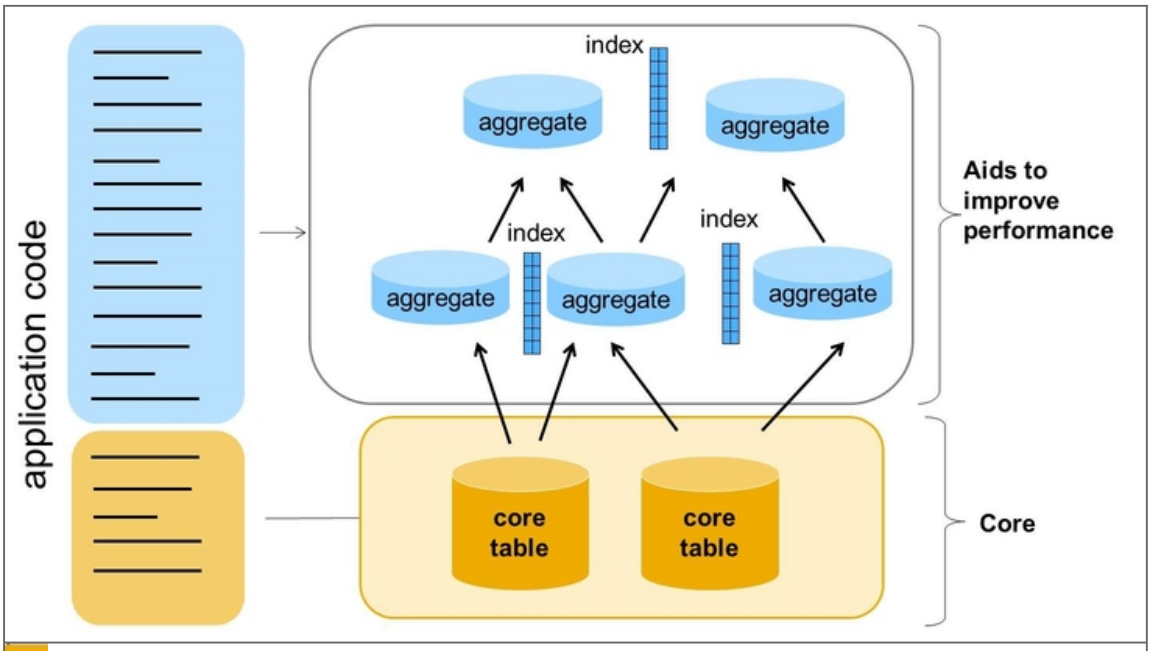


Figure 7: Traditional Approach to Data Model and Application Code

Traditional applications were built on a hierarchical data model. Detailed data was summarized into higher level layers of aggregates to help system performance. On top of aggregates, we built more aggregates and special versions of the database tables to support special applications. As well as storing the extra copies of data, we also had to build application code to maintain extra tables and keep it up to date. A backup to these extra tables was also required, so even the IT operations were impacted.

In addition to aggregates, we have another inefficiency that we need to remove. Database indexes improve access speed as they are based on common access paths to data. However, they need to be constantly dropped and rebuilt each time the tables are updated. So again, more code is needed to manage this process.

The traditional data model is complex and this causes the application code to be complex. 70% of application code is built specifically for performance of an application and adds no value to the core business function.

A complex data model and complex code, means that integration with other applications, and also enhancements, are difficult. This means that they are simply not agile enough for today's fast moving business environment.

Eliminate Complexity of Applications with SAP HANA

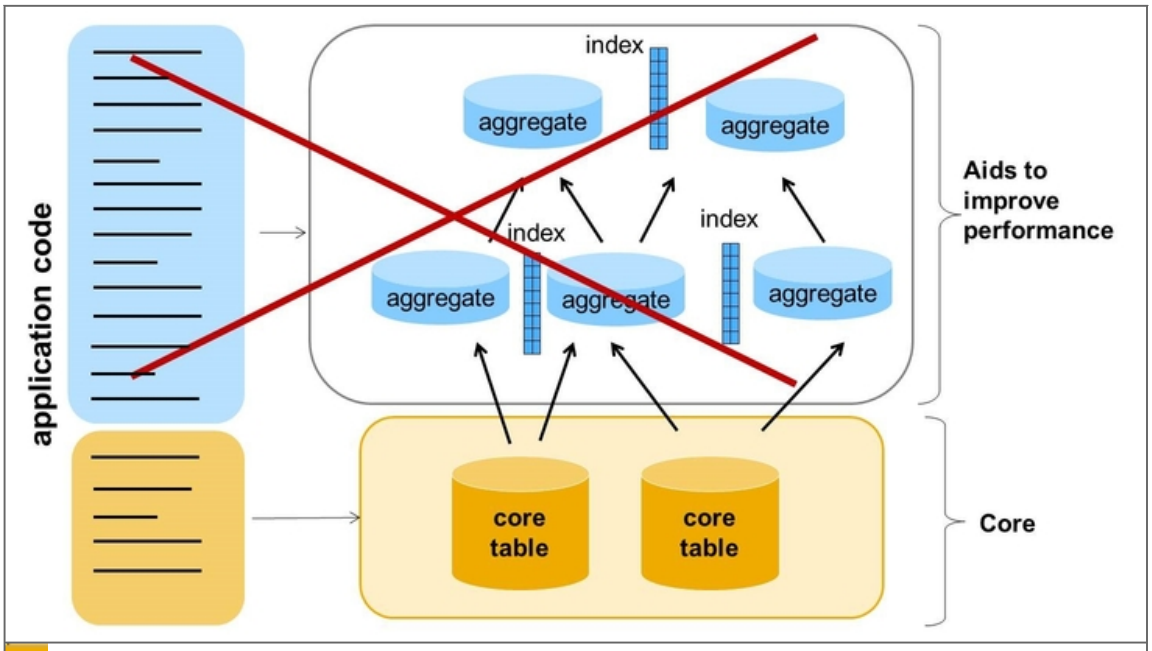


Figure 8: Eliminate Complexity of Applications with SAP HANA

Using the power of SAP HANA, you can aggregate on the fly from any line item table. You do not need prebuilt aggregates. SAP HANA can generate any view of the data at run time, all from the same source tables.

SAP HANA organizes data using column stores, which means that indexes are usually not needed. They can still be created but offer little improvement.

As well as removing the aggregates and indexes from the database, you can also remove huge amounts of application code that deals with aggregates and indexes.

You are left with a simplified core data model as well as simplified application code.

Now it is much easier to enhance the applications and integrate additional functions.

Movement of Processing to the Data

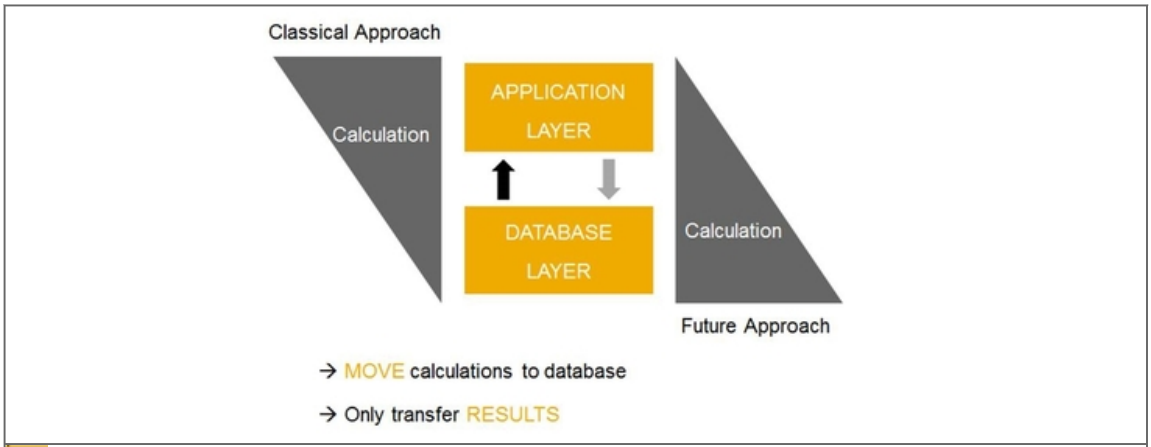


Figure 9: Push Down Processing to SAP HANA

In the past, the role of the database layer was to listen out for instructions from the application layer and act upon these instructions. These instructions could be any of the following:

- Create data
- Read data
- Update data
- Delete data

These requests were simple and required little or no actual processing or calculating of the data. For example, in the case of reading data, the application layer would take care of the data processing tasks once the database had “handed over” the basic data. The database was simply told to fetch large blocks of data, or write large blocks of data, and so on. However, SAP HANA is much more than a database, it is a data processing platform.

This means that SAP HANA doesn't just want to be given simple data tasks, it wants to be given all data centric jobs. The more complex the task, the better. These jobs can include the following:

- Aggregate and disaggregate data
- Filter and sort data
- Calculate data
- Convert data
- Cleanse data
- Structure data

Push Down Process

So, an application should now send all data instructions, whether simple or complex, to SAP HANA. SAP HANA processes the data in memory and sends back only the results. This means that often, less data passes between the database and application layer.

For example, if the application layer sends the instruction “please summarize the last 5 years’ sales line items of yellow widgets into region totals by year, and calculate the net value after discount” to SAP HANA. Instead of sending millions of basic rows from the database to the application layer, SAP HANA processes the data request and sends back only the results to the application layer. A huge reduction in data volume is being passed. As well as this, the data processing is done in memory by SAP HANA, so performance is excellent too.

Moving the data processing tasks from the application layer to the database layer is called push-down. Push-down means that application developers need to rethink their approach. In the past, all coding focused on the application layer, but now with SAP HANA, large parts of the coding can be pushed down. This means that developers need to think of how to ensure that they pass challenging data processing tasks to SAP HANA, instead of expecting the application layer to handle this.

For simple applications it is possible that SAP HANA can take care of all processing, and a separate application server is not needed. SAP HANA has a built-in application server called SAP HANA XS, which we will cover in detail later in the course.

However, the application layer is still needed with complex enterprise applications such as SAP S/4HANA and Business Warehouse (BW). It is needed to handle the complex business logic that must be programmed in a dedicated business programming language. In the case

of SAP applications, the language is ABAP. You cannot develop and run ABAP applications directly in SAP HANA. For that, you need SAP NetWeaver Audience Sensor ABAP, which sits on top of SAP HANA. Now you have two platforms working as an optimized stack – SAP NetWeaver Audience Sensor ABAP, which provides the application services, running on top of SAP HANA, which provides the data services.

Data Access from Anywhere

In today's digital world there are many types of data. As well as the traditional business data of structured records, we also have the following data types:

- Text Data

  This is data from social media feeds, help desk tickets, logs, and so on.

- Spatial Data

  This is data that relates to locality, maps, engineering diagrams, floor plans, and so on.

- Graph Data

  This is data that relates to highly networked entities such as social networks, supply chains, and so on.

Access Any Type of Data

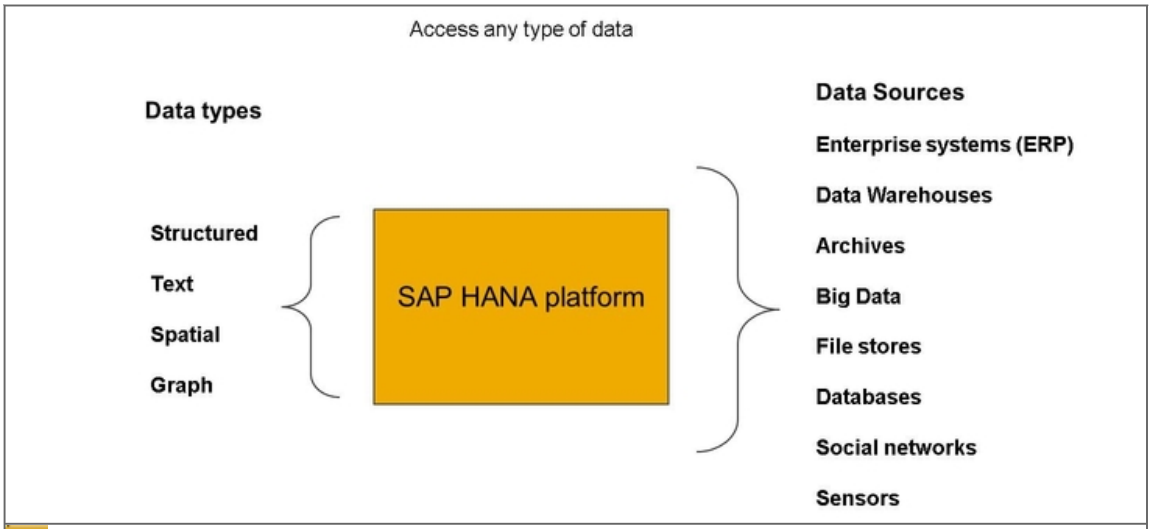


Figure 10: Access Any Type of Data

SAP HANA can process all types of data and also combine them in new and innovative applications. Imagine a cockpit that provides a summary of customer feedback based on an aggregation of social media comments. Imagine being able to drill down on any customer sentiment to see if there is any regional aspect to the sentiment. Finally, imagine drilling down on the map to open up all sales orders for that region related to the original sentiment.

SAP HANA can also access data from any type of source system, including the following:

- Enterprise Systems

  S/4HANA is an example of an enterprise system

- Data Warehouses

  BW is an example of a data warehouse

- Archives

  SAP HANA can create an online connection to remote archives

- Big Data

  Apache Hadoop is an example of big data

- File Stores

  XL, CSV, and XML are examples of file stores

- Databases

  This includes any relational database

- Social Networks

  Twitter, Facebook, and LinkedIn are examples of social networks

- Sensors

  These are embedded databases and data containers in smart devices and machines

Access Data Anywhere

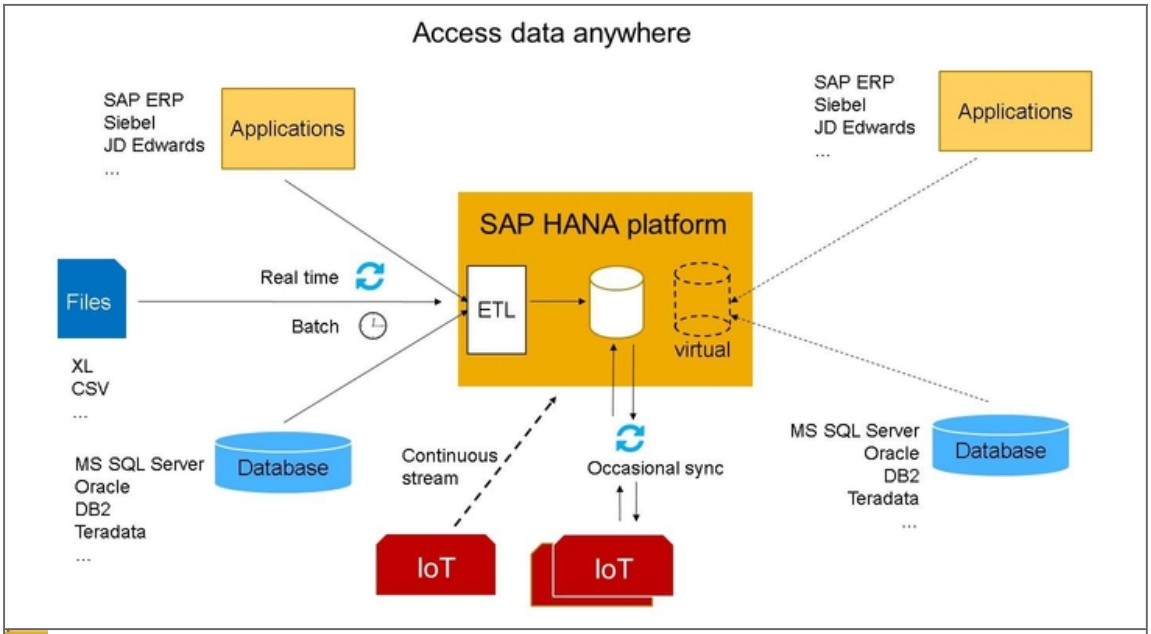


Figure 11: Access Data Anywhere

SAP HANA can consume data for processing from anywhere.

Data can be physically loaded to SAP HANA either in real time or batch. Data can be accessed virtually from remote sources to obtain a live view of data without loading. Data can be loaded from files, databases, continual streams, or occasionally connected sources.

SAP HANA has its own built-in ETL tools. This means that integrating and cleansing data is possible without the need to implement other tools. On the other hand, if external data provisioning tools are required, then SAP HANA can natively connect to those too.

Platform Services to Power any Type of Application

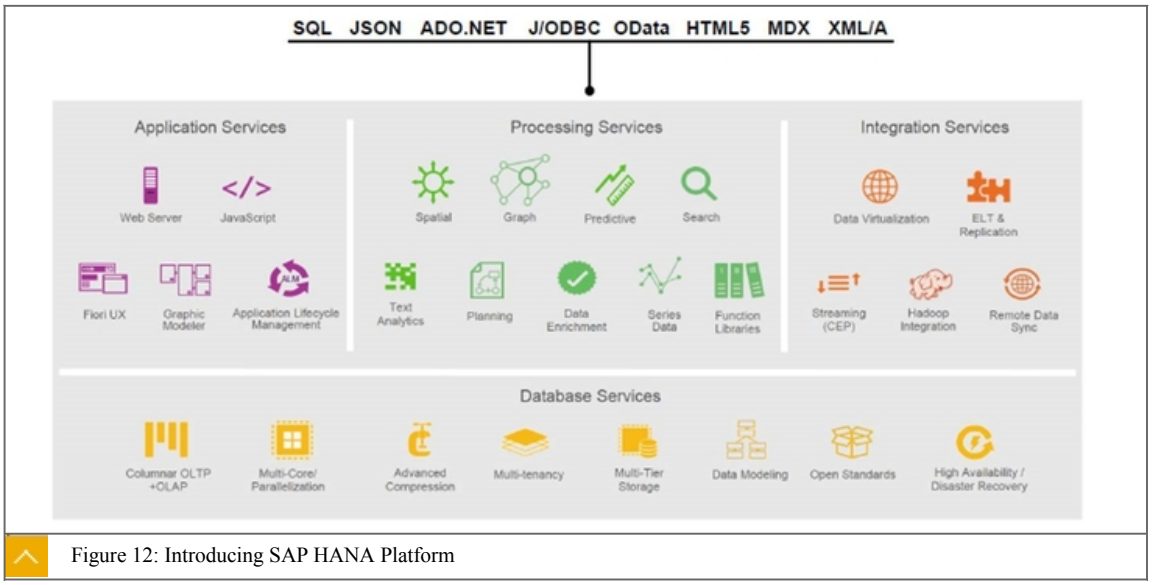


Figure 12: Introducing SAP HANA Platform

SAP HANA is a platform. A platform is a combination of many services that power applications. A platform should include all services required by the applications. The platform is the software that was built entirely by SAP only to run on the most powerful hardware.

The SAP HANA platform combines all key components that can be used to power any application. These components include database, data processing, application development tooling, lifecycle management, and data integration. They can be used in both SAP and non-SAP applications.

Service Key Capabilities of SAP HANA

The following are some of the key capabilities of SAP HANA, organized by service:

- Application Services

  As well as a database, SAP HANA also provides many application services. This means many applications are built in a two tier model, rather than a three tier model. For example, imagine an application that allows a project manager to quickly check that all team members have competed their time sheets. This could easily be developed as a web application where only a web browser and SAP HANA is required, and no application server is needed. This is because SAP HANA can handle the business logic as well as the database services. SAP HANA provides a full development environment with productivity tool supplied in the box. Everything the developer needs at design time and run time is there.

- Processing Services

  SAP HANA can handle many new types of data. This includes text, spatial, graphic, and more. However, it is not enough to simply store these new data types. You also need to be able to build applications that can process and integrate this data with traditional data types, such as business transactions. SAP HANA provides native in-memory engines that process all types of data in real time.

- Integration Services

  SAP HANA has the following built-in data consumption options:

- Continual streaming data analysis
- Read data remotely in any data source
- Read from Big Data stores such as Hadoop
- Synchronizes in both directions with remote databases and devices that collect data (IoT)

SAP HANA has built-in Extraction, Transformation, and Loading (ETL) capabilities. This means that separate software is no longer needed to clean, enrich, and profile data from any source.

- Database Services

  SAP HANA is a full, in-memory column-and-row store database that can support both OLTP and OLAP requirements and is built to run on high-end hardware. It stores data optimally using automatic compression and is able to manage data on different storage tiers to support data aging strategies. It has built-in high-availability functions that keep the database running and ensure mission-critical applications are never down.

Partnership of Hardware and Software

In this concept you learn about some basic technical aspects of SAP HANA and the choices available to customers.

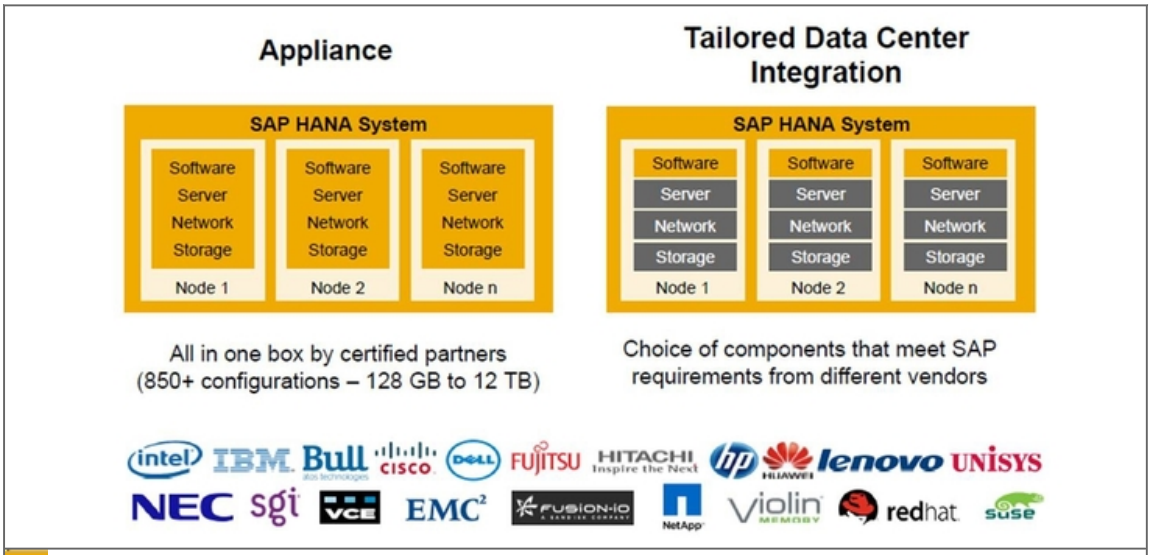


Figure 13: Configuration Choices

For on-premise deployments, SAP HANA can be delivered as an appliance. This means the SAP HANA software is pre-installed on certified hardware. There are many suppliers who can provide hardware that is certified by SAP to run SAP HANA optimally. Customers work with the hardware supplier to size the correct hardware and to choose the various configuration options.

Alternatively, SAP HANA can be installed flexibly on mixed hardware components that appear on SAP's approved hardware list. Many customers already have hardware components as well as software licenses that they would like to repurpose. So, this flexible approach ensures that implementation costs are kept to a minimum and that hardware is recycled. This approach is known as Tailored Data Center Integration.

Note:
Only certified SAP HANA engineers are allowed to install SAP HANA for production purposes. This restriction does not apply for non-production installations.

Versions of Linux

SAP HANA runs on the Linux operating system. The following versions of Linux are supported:

- SUSE
- Red Hat

SAP HANA runs on either the Intel x86 chipset, or the IBM Power chipset.

SAP HANA in the Cloud

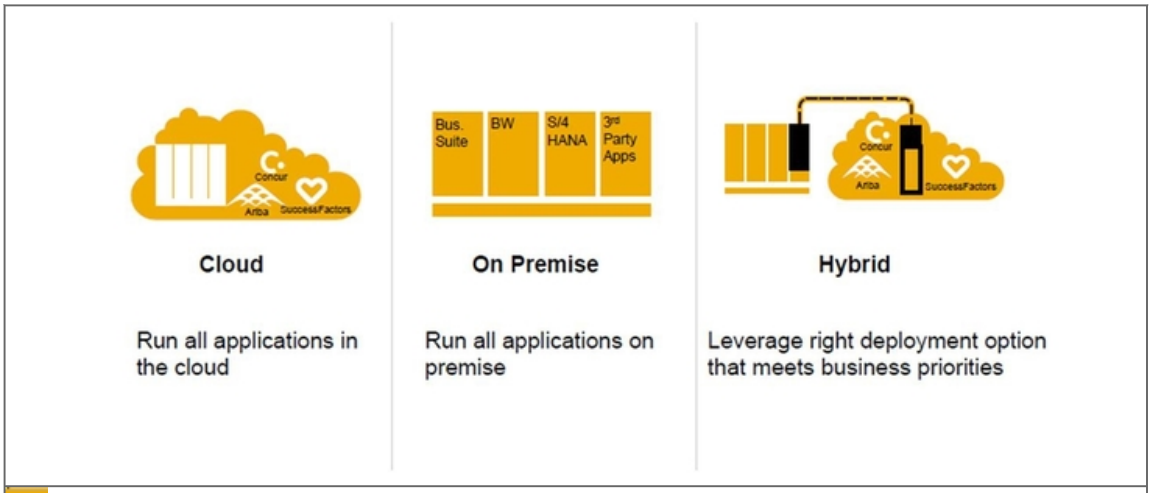


Figure 14: Flexible Deployment Options

SAP HANA is available on-premise as well as in the cloud.

On-premise means that the entire solution, the software, network, and hardware is installed and managed by the customer. The advantage of the on-premise option is that customers have complete control over maintenance and upgrades. The disadvantage is that customers must have their own in-house resources, and purchase or lease their own hardware and licenses.

Cloud Deployment Approach

A cloud deployment is operated and managed by SAP and other hosting partners. The advantage of the cloud option is that customers do not have to be concerned with providing and managing the infrastructure. They do not need to have their own operation resources to run SAP HANA. They can simply get on with using and developing applications with SAP HANA. It also means customers do not have to purchase hardware and software licenses. SAP HANA is paid for by subscription. It is considered to be a service. The disadvantage with the cloud is that SAP and partners control the maintenance and upgrade schedules, so customers have no control over this. Thus, they have to follow SAP's lifecycle, which tends to move forward at a good pace. This might not be ideal if the customer prefers to control their own lifecycle, especially if the customer is constantly developing add-on applications and needs a stable platform.

Note:
Be careful not to confuse a public cloud offering with managed cloud. Both are cloud solutions. SAP HANA Public Cloud is shared by many customers. The cloud-service provider supplies SAP HANA, the hardware, and the operating resources. A managed cloud is a customer's own private SAP HANA platform. It is managed by SAP or hosting partners who provide the hardware and operate the system, but the customer must provide SAP HANA and applications.

Hybrid Approach

Another option is a hybrid approach, where a combination of on-premise and cloud is used. For example, a customer wants to have an on-premise deployment of SAP HANA to run their ERP solution. However, they are also developing new applications that require more complex SAP HANA services and infrastructure than they currently have in-house. Because of this, they use a cloud version of SAP HANA to run these applications. A key point to remember is that a hybrid solution brings together applications in the cloud and on-premise, and there should be no barriers to developing applications that cross both. It is possible to develop applications in SAP HANA that can easily be swapped between on-premise and cloud (and the other way around). These are called multi-target applications (MTA).

SAP HANA Scenarios

SAP HANA is central to SAP's strategy of providing a next-generation digital platform that can power both existing and new applications. These can be either SAP or non-SAP applications.

SAP HANA can handle any type of application, including analytical, transactional, consumer facing, back-office, real-time, predictive, cloud, and more. With so much versatility, SAP HANA aims to be the single platform for an organization.

So let's take a quick look at some examples of where SAP HANA can be used.

Scenarios for SAP HANA

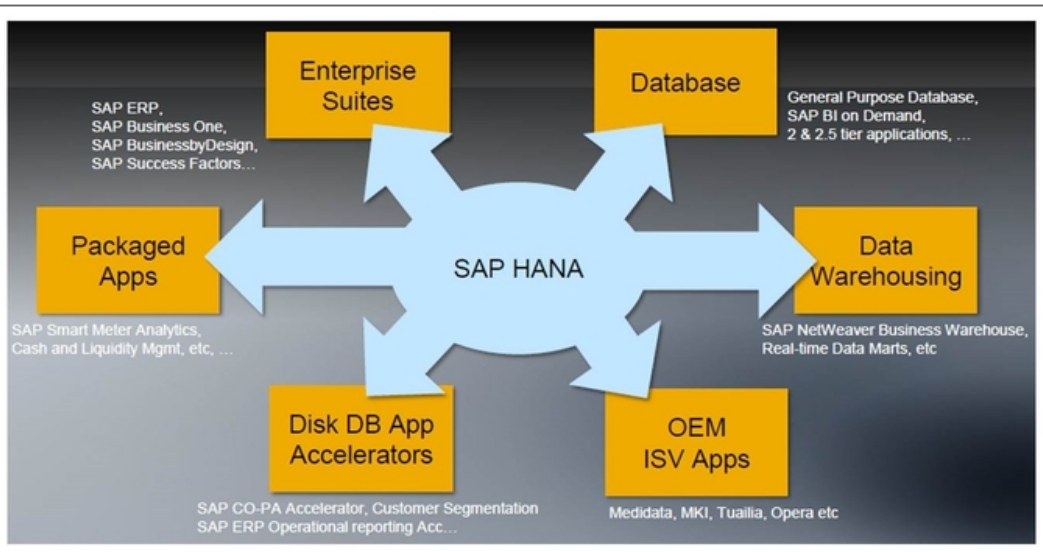


Figure 15: Scenarios for SAP HANA

There are many use cases and scenarios for SAP HANA. These include the following:

- To power enterprise suites such an S/4HANA and Business Suite
- To power data warehouses such as BW, or use it to build local data marts
- Wherever a relational database is needed
- As an embedded data processing engine and data store in partners' applications
- To work side by side with traditional applications to accelerate business processes that are data-processing rich, such as MRP or customer segmentation
- To power innovative line of business or vertical market packaged applications that are built to run on SAP HANA, such as Sport Analytics or Smart Meter Analytics

Powering Existing Applications with SAP HANA

For many customers, their first experience of SAP HANA will be when they replace the traditional databases running their existing SAP applications with SAP HANA. For many years SAP applications such as SAP Business Suite, SAP BW, SAP CRM, and more have been running on traditional disk-based databases. These disk-based databases include IBM, Microsoft, and Oracle, among others.

Migrating these legacy databases to SAP HANA is easy, and in fact SAP provides all the tools to automate the migration. When SAP powers existing applications, tremendous performance improvements can be expected. However, it is not just performance that is improved. SAP HANA provides many additional opportunities to expand the scope of existing applications. Examples of this potential expanded scope include the following:

- Embedding textual analytics into existing ERP processes to find out what customers are saying about your service
- Combining sensor data (Big Data) with BW data to build a super warehouse of data for drill-down at any level

Powering SAP Enterprise Applications with SAP HANA

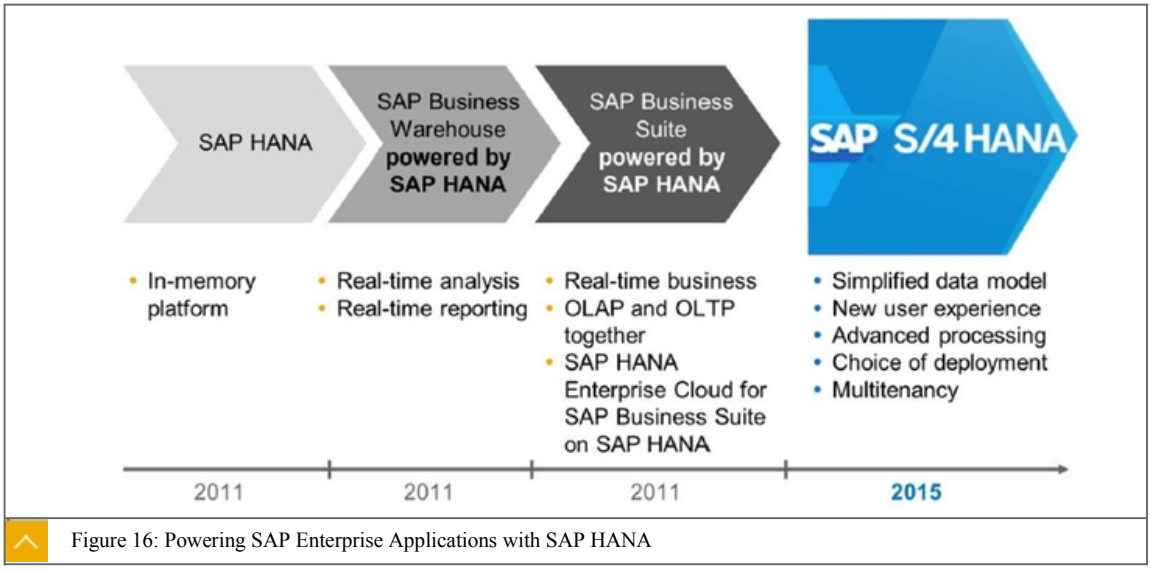


Figure 16: Powering SAP Enterprise Applications with SAP HANA

Next Generation Suite

SAP HANA was released in 2011 and at that time it did not power any SAP applications. Its main use case was as an analytical engine to power Business Intelligence. Many customers

implemented SAP HANA and loaded data from the ERP systems, either in batch or real-time, so that they could build BusinessObject reports on top of SAP HANA.

However, not long after its introduction, SAP HANA was able to completely replace the legacy databases running under BW. Following the success of BW on HANA, SAP HANA could then be used to replace the legacy databases running under Business Suite. Thus, today there are many customers running BW or Business Suite on HANA, whose first encounter with SAP HANA was following a database migration. Customers implementing SAP S/4HANA meet SAP HANA right away.

SAP HANA is the only database that powers this next generation suite. It combines exceptional performance and simplicity, which gives you massive opportunities to grow your digital business.

S/4HANA never ran on legacy databases.

Powering New Applications with SAP HANA

SAP HANA is not just for existing SAP applications or even just enterprise suites. SAP HANA is used to power many other exciting and innovative applications in all types of environments. SAP HANA can be used whenever massive processing power is needed for large volumes of data of any types.

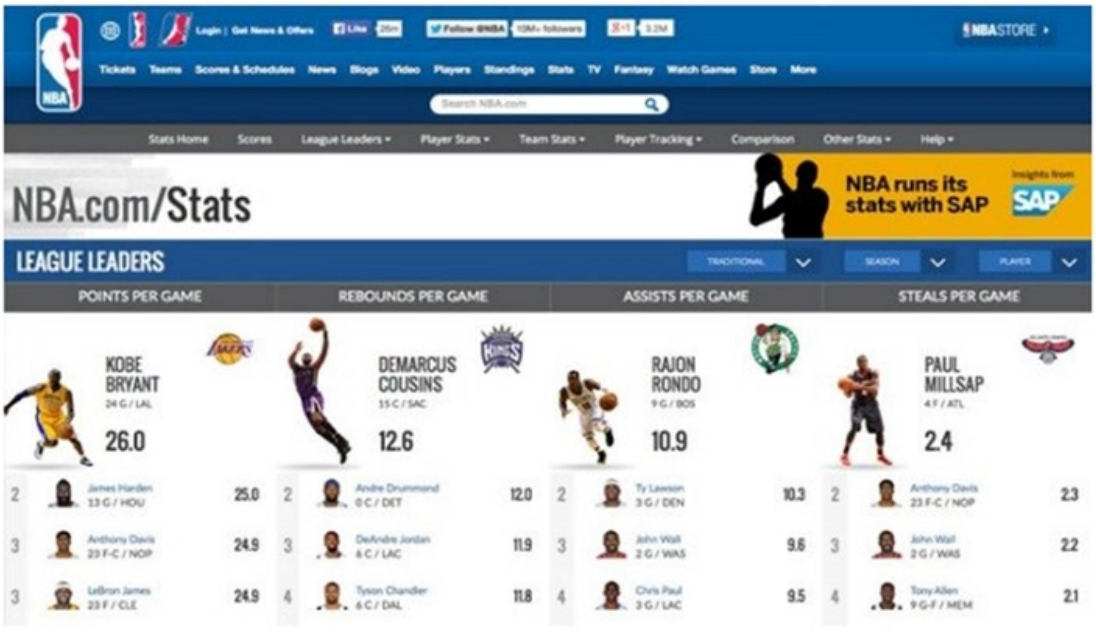


Figure 17: Powering New Applications with SAP HANA

Sports Analytics is a good example of a new, innovative application powered by SAP HANA. It provides real-time, in-game analytics to sports fans. For example, player and team statistics can be analyzed to enhance the fan experience. Sports coaches also use SAP HANA powered applications to improve the performance of their teams. Sensors in the ball and on other equipment provide real-time analysis of play.

Choice of SAP HANA Versions

SAP HANA is available in two versions; SAP HANA 1.0 (launched in 2011), and SAP HANA 2.0 (launched in 2016). Although you may assume that SAP HANA 1.0 is now out of date, in fact both versions are relevant and each has its own use case.

Since the launch of SAP HANA 1.0, the product has evolved at a phenomenal pace with new features and capabilities being added continuously. SAP has been keen to make available the new features as soon as possible. So SAP delivered updates to SAP HANA through support packs stacks (SPS) every six months. However, customers who run mission critical applications (e.g. S/4HANA) prefer an SAP HANA platform with less frequent updates, thus avoiding working through an upgrade project so often. They may also not require the new functionality delivered in each support pack and thus are not motivated to upgrade. However, in parallel, those customers may also be developing new applications and would like to take advantage of the latest innovations. So SAP decided to offer a dual-track approach to satisfy both needs.

SAP HANA: Versions

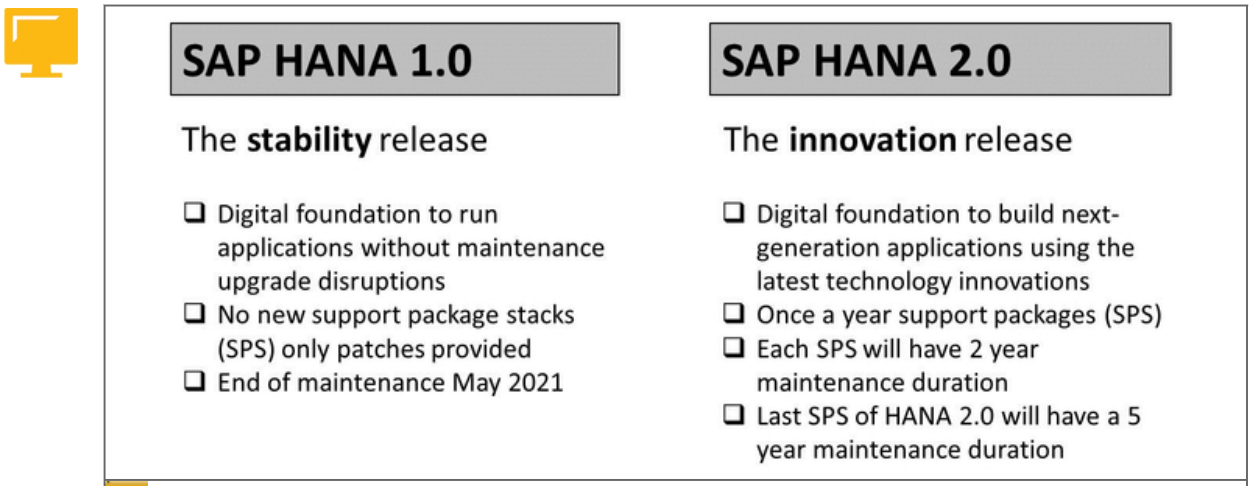


Figure 18: Two Versions of SAP HANA

So SAP HANA 1.0 SPS12 is ideal for customers seeking a stable platform without frequent updates to run their mission-critical applications. There will be no support packs after SPS12 on SAP HANA 1.0. Only software fixes are provided, through maintenance revisions, until May 2021. Customers running SAP HANA 1.0 should consider upgrading to SPS12 if they are on lower support packs, to get the maintenance until May 2021.

SAP HANA 2.0 is ideal for customers seeking a platform that enables them to leverage the very latest technology innovations. Customers can build next-generation applications and also run mission-critical applications on SAP HANA 2.0. The SAP HANA 2.0 platform inherits all the features and capabilities of SAP HANA 1.0 but with significant additional innovations. All content built with SAP HANA 1.0 continues to run on SAP HANA 2.0 with no changes.

Upgrade Paths for Existing Customers

Customers running SAP HANA 1.0 SPS10 – SPS12 can upgrade directly to SAP HANA 2.0. However, SAP recommends that those customers first take advantage of the new “Capture and Replay” tool, introduced for SPS12. This tool allows you to capture existing workloads from SPS12 and see how they run on HANA 2.0 This can help to identify any configuration issues, allowing customers to fine tune the target HANA 2.0 system before committing to the changeover.

Customers running SAP HANA SPS09 and lower must first upgrade to SAP HANA 1.0 SPS12 in order to upgrade to SAP HANA 2.0.

Unit 1
Exercise 1

Set Up the Training Environment

The purpose of this exercise is to set up the training environment so that all future exercises in this course can be executed. Some exercises require ready-made files. You will copy these files to your local work area by running a prepared script so they become available to you.

1. Log on to your training landscape.

> Note:
> The logon procedure to the training landscape will be provided by your instructor or included in the SAP Live Access instructions.

2. (Live Access users only) Connect to your own Live Access system as indicated in the SAP Live Access general user guide.
3. (Classroom Training, Virtual Live Classroom only) Start a remote desktop connection to the training landscape using the following data:

Table 1: Connect to Remote Desktop

Field	Value
Computer	<provided by your instructor >
User Name	**`train-##`**
Password	**`initial`**

If a dialog box displays the message, The identity of the remote computer cannot be verified... , select the checkbox Don’t ask me again and choose Yes .

4. Run the HA100 initialize script so that the course folder HA100 and its contents is copied to your local environment, in N:\HA100 .
5. Add the course folder N:\HA100 to your Favorites so it is easy to access in future.
6. Check that the course resources are present.

Unit 1
Solution 1

Set Up the Training Environment

The purpose of this exercise is to set up the training environment so that all future exercises in this course can be executed. Some exercises require ready-made files. You will copy these files to your local work area by running a prepared script so they become available to you.

1. Log on to your training landscape.

> Note:
> The logon procedure to the training landscape will be provided by your instructor or included in the SAP Live Access instructions.

2. (Live Access users only) Connect to your own Live Access system as indicated in the SAP Live Access general user guide.

3. (Classroom Training, Virtual Live Classroom only) Start a remote desktop connection to the training landscape using the following data:

   Table 1: Connect to Remote Desktop

Field	Value
Computer	<provided by your instructor >
User Name	**`train-##`**
Password	**`initial`**

   If a dialog box displays the message, The identity of the remote computer cannot be verified... , select the checkbox Don’t ask me again and choose Yes .

   a) On the SAP Training Landscape portal or Common Training, choose the Remote Desktop Connection icon.

   b) In the Remote Desktop Connection dialog box, in the Computer field, enter the Remote Computer name provided by your instructor and choose Connect .

   c) In the Windows Security dialog box, choose Use another account or for Windows 10, select more choices > Use a different account. .

   d) Enter the user name and password from the table, Connect to Remote Desktop, and choose OK.

4. Run the HA100 initialize script so that the course folder HA100 and its contents is copied to your local environment, in N:\HA100 .

   a) In the Remote Desktop environment, choose the Start button.

   b) Select the Initialize Course tile.

c) Double-click the HA100 folder.

d) To execute the script, double-click the Initialize_HA100 shortcut.

e) At the next prompt, choose Yes to execute the script.

5. Add the course folder N:\HA100 to your Favorites so it is easy to access in future.

a) To create a link, drag the HA100 folder from the local N:\ drive to Favorites .

6. Check that the course resources are present.

a) In Favorites , choose the HA100 folder and ensure that various ready-made files and shortcuts are present.

LESSON SUMMARY
You should now be able to:

- Describe how SAP HANA powers a digital platform

Unit 1

Learning Assessment

1. Which recent trends have triggered the need for a next generation data processing platform?

   Choose the correct answers.

   - ☐ **A** Increasing connectivity of people and devices
   - ☐ **B** Move transactional processing and analysis to separate servers for better scalability
   - ☐ **C** Increase in use of mobile devices
   - ☐ **D** Massive growth in data volume

2. Which recent technology innovations have triggered the opportunity to build a next generation data processing platform?

   Choose the correct answers.

   - ☐ **A** Faster disk access
   - ☐ **B** Multi-core processing
   - ☐ **C** Larger memory availability

3. What are the key features of SAP HANA?

   Choose the correct answers.

   - ☐ **A** In-memory database
   - ☐ **B** Automatic compression of data
   - ☐ **C** Removes the need for cache
   - ☐ **D** Combined OLTP and OLAP processing on one platform

4. Which are true statements?

   Choose the correct answers.

   - ☐ **A** SAP HANA runs on Intel x86 and SPARC hardware platforms.
   - ☐ **B** SAP HANA runs on Unix.
   - ☐ **C** SAP HANA can be installed only by certified engineers for production purposes.
   - ☐ **D** Customers can re-use their own certified hardware components in an SAP HANA system.
   - ☐ **E** SAP HANA is available in the cloud or on premise, or a combination of both.

5. SAP HANA 2.0 is the ‘stability’ release and its use case is to support mission critical applications where frequent upgrades are not desirable.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

Unit 1

Learning Assessment - Answers

1. Which recent trends have triggered the need for a next generation data processing platform?

   Choose the correct answers.

   [X] **A** Increasing connectivity of people and devices

   [ ] **B** Move transactional processing and analysis to separate servers for better scalability

   [X] **C** Increase in use of mobile devices

   [X] **D** Massive growth in data volume

   You are correct! Transactional processing and analysis are now coming back together to support a simpler landscape and enable real time decision making removing the need to move the data from one system to another: Refer to HA100 Unit 1 Describing SAP HANA for details

2. Which recent technology innovations have triggered the opportunity to build a next generation data processing platform?

   Choose the correct answers.

   [ ] **A** Faster disk access

   [X] **B** Multi-core processing

   [X] **C** Larger memory availability

   You are correct! We are moving away from disk and towards memory based data storage and processing using multi-core processors: Refer to HA100 Unit 1 Describing SAP HANA for details

3. What are the key features of SAP HANA?

   Choose the correct answers.

   [X] **A** In-memory database

   [X] **B** Automatic compression of data

   [ ] **C** Removes the need for cache

   [X] **D** Combined OLTP and OLAP processing on one platform

   You are correct! SAP HANA uses multi-level cache in order to provide excellent performance: Refer to Unit 1 Describing SAP HANA for details.

4. Which are true statements?

   Choose the correct answers.

   [ ] **A** SAP HANA runs on Intel x86 and SPARC hardware platforms.

   [ ] **B** SAP HANA runs on Unix.

   [X] **C** SAP HANA can be installed only by certified engineers for production purposes.

   [X] **D** Customers can re-use their own certified hardware components in an SAP HANA system.

   [X] **E** SAP HANA is available in the cloud or on premise, or a combination of both.

   You are correct! SAP HANA runs only on Intel x86 and IBM Power Series platforms. SAP HANA runs on Linux not Unix: Refer to Unit 1 Describing SAP HANA for details.

5. SAP HANA 2.0 is the ‘stability’ release and its use case is to support mission critical applications where frequent upgrades are not desirable.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! SAP HANA 1.0 is the ‘stability’ release with no new features, only maintenance patches, being offered. SAP HANA 2.0 is the ‘innovation’ release and receives upgrades every 6 months with significant new features added: Refer to Unit 1 Describing SAP HANA for details.

UNIT 2 Architecture of SAP HANA

UNIT OBJECTIVES

- Outline components of SAP HANA
- Work with SAP HANA interfaces
- Understand key features of SAP HANA database
- Understand run-time and design-time architectures
- Describe high availability
- Describe security features of SAP HANA

Unit 2
Lesson 1

Outlining Components of SAP HANA

LESSON OVERVIEW
This lesson provides an overview of the key components in an SAP HANA landscape.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Outline components of SAP HANA

Core SAP HANA Components

An SAP HANA landscape is made up of many components. Not all components are required by customers and so rather than delivering one super-sized version of SAP HANA containing all components – therefore requiring more hardware and resources – SAP provides different **editions** of SAP HANA. It might help to think of editions as 'bundles' of components. The current SAP HANA editions are as follows:

SAP HANA, enterprise edition
SAP HANA, platform edition
SAP HANA, spatial
SAP HANA, base edition

The base edition contains only the essential components, whereas the enterprise edition is fully loaded.

Note:
You can find out more about the latest editions and what they contain by going to the SAP Help Portal and looking up the **Feature Scope Description for SAP HANA** .

Some components are fundamental to the running of SAP HANA and are included in all editions. As well as editions, SAP HANA features are available as **options** . Options provide individual components that can be installed to SAP HANA. It might help to think of options as a menu of items you can chose from to add to your edition.

In this lesson we don't focus on editions and options but describe SAP HANA without such constraints. We will first examine a simple SAP HANA landscape and then later we will expand this to describe more components.

SAP HANA Basic Landscape

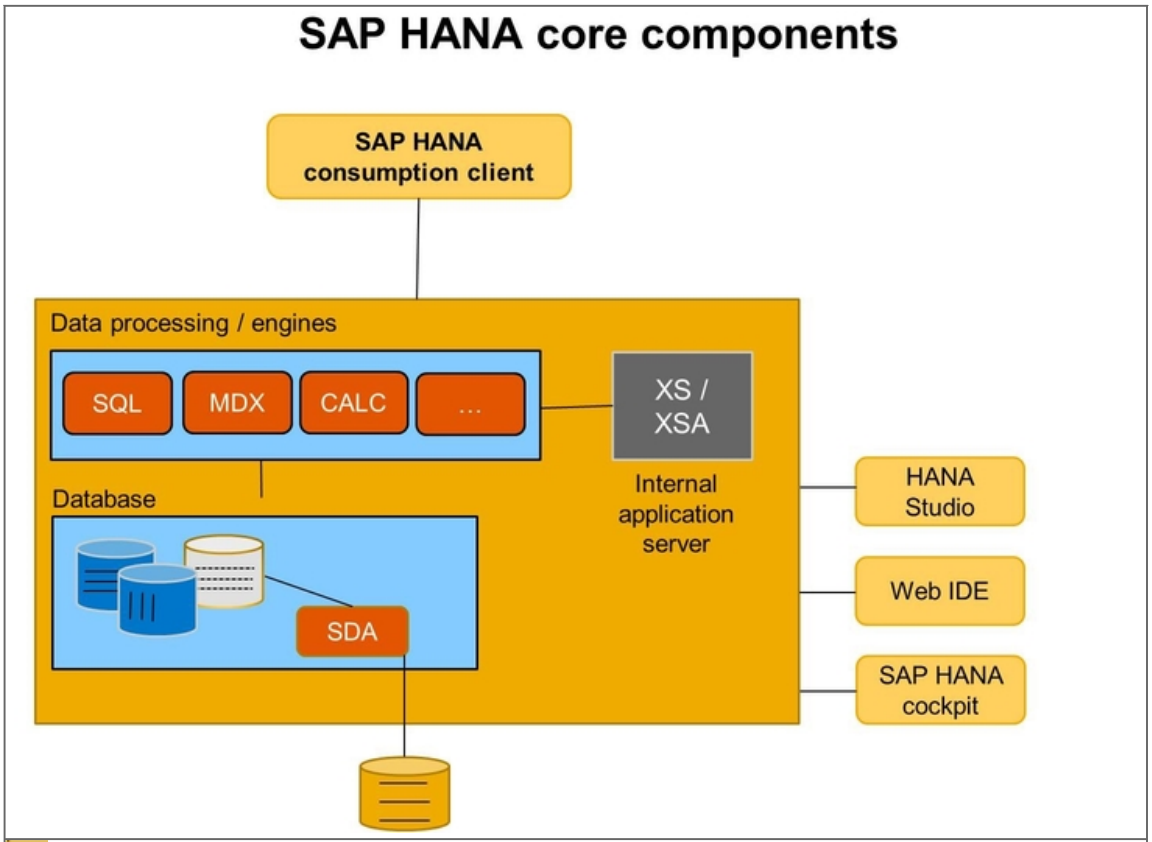


Figure 19: SAP HANA Core Components

At the heart of SAP HANA is the in-memory database. The SAP HANA database uses row and column store tables. The database is a fundamental component of SAP HANA and therefore it is a mandatory part of the landscape.

The database tables can contain physical data. They can also be virtual tables that are connected to external data sources which provide live data on request. A virtual table is a logical table that contains no data in SAP HANA but it is treated just like a regular table. The Smart Data Access (SDA) component performs the provision of live data to the virtual tables.

Data Processing Engines

Another set of essential components of SAP HANA are the data processing engines. These dedicated engines provide the in-memory number crunching capabilities. They listen out for requests from applications and then interact closely with the database to produce fast results within the database. There are many engines available, each specializing in handling different types of data. But the engines work together, and special optimizers figure out which engine should be called to get the best performance.

SAP HANA provides relational access to the database via SQL and multidimensional access via MDX. There are dedicated processors to handle these query requests. The calculation engine is able to undertake complex analytical tasks, such as hierarchical and deep OLAP queries. It works in harmony with SQL and MDX processors to produce the fastest possible results.

XS/XSA

Extended Application Services (XS) and its more powerful replacement, Extended Application Services – Advanced (XSA) are the two built-in application servers that are part of core SAP HANA. With XS or XSA, you have all the development and runtime tools needed to develop your own native SAP HANA applications and run them with no separate application server required (e.g. NetWeaver). All you need to add is a front-end client such as a browser and you have the complete end-to-end application stack.

Interfaces

The SAP HANA Studio is a thick client (runs on Windows, MacOS, or Linux) and provides the front-end interface that serves the needs of multiple roles including developers, data modelers, and administrators. These people log on directly to SAP HANA to access the development tools or perform administration duties. With SAP HANA Studio you can only develop XS based applications (not XSA).

The Web IDE for SAP HANA is a powerful web interface used to develop XSA based applications and data models. It cannot be used for XS development. The Web IDE for SAP HANA contains no administration tools, it is purely a developers interface.

The SAP HANA Cockpit is used for administration and monitoring of the SAP HANA platform and is a modern SAP Fiori based web cockpit.

Extended SAP HANA Landscape

As well as the core components, there are many additional components that make up an SAP HANA landscape. Customers choose these additional components based on their needs.

Note:
This is not a complete list of all possible additional components, but is meant to provide an illustration of the type of components that are available.

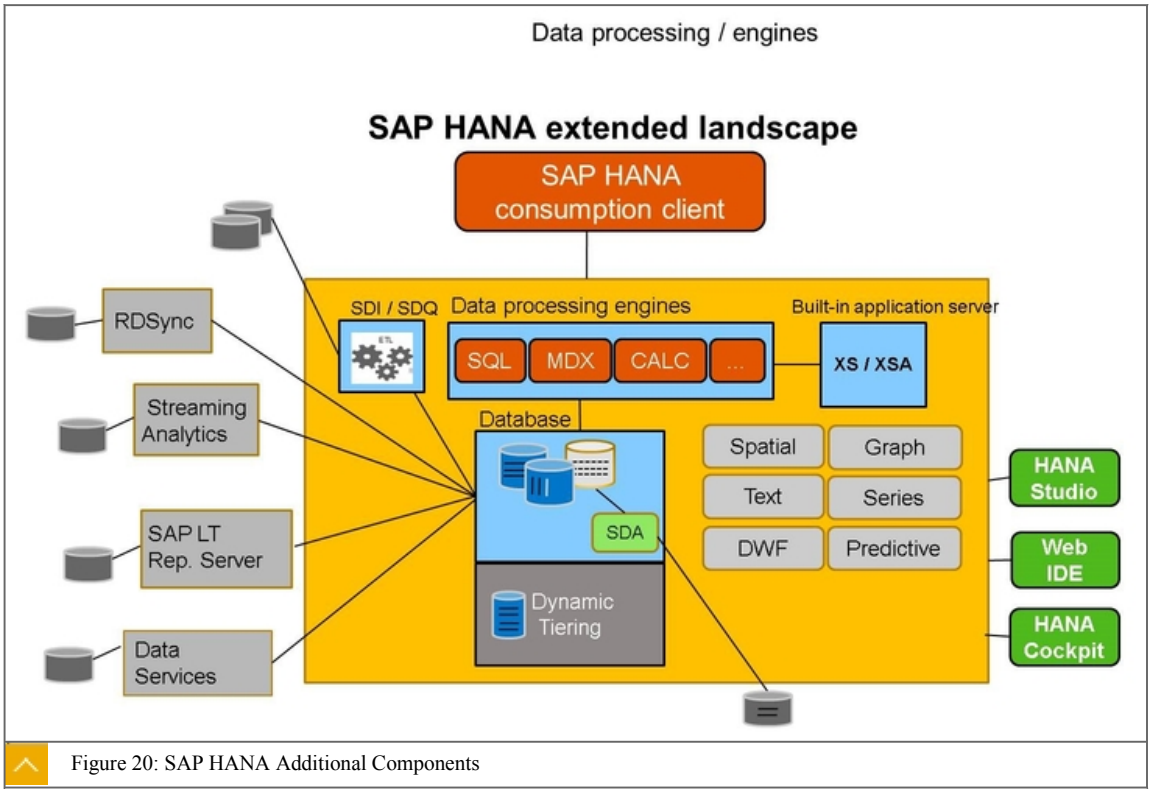


Figure 20: SAP HANA Additional Components

SDI/SDQ

SAP HANA has its own built-in Enterprise Information Management (EIM) component that takes care of the majority of data provisioning scenarios. This is used to provision data from any sources to SAP HANA in real-time or batch. EIM consists of a component called Smart Data Integration (SDI). SDI takes care of extraction, transformation, and loading (ETL) of data. The other component of EIM is Smart Data Quality (SDQ). You use SDQ to enrich and cleanse data during the provisioning process.

SAP Landscape Transformation Replication Server and SAP Data Services

SAP HANA natively connects to other SAP data provisioning tools, such as SAP Landscape Transformation Replication Server (SAP LT Replication Server) and SAP Data Services. These optional tools are used to acquire and transform data from SAP and non-SAP sources in real-time or batch. Many customer already use these tools, and so SAP have ensured that they are able to work natively with SAP HANA. But SAP recommends the use of the built-in EIM component in order to simplify the SAP HANA landscape, so these external tools can be mostly eliminated.

Streaming Analytics and Remote Data Sync

Streaming Analytics (previously known as SDS) is a component that allows you to connect SAP HANA to live continual multi-channel streams of data. These could be from connected machines or sensors or logs.

RDSync is the remote data synchronization component that is used to connect to enterprise databases and mobile device databases that cannot be online continuously. It provides two-way synchronization services.

Advanced Data Processing

Text processing is an additional component that can be added to provide text processing and analysis services, such as text search and text mining.

Spatial processing is another additional component that can be added to provide analysis services. These are based on spatial data, such as queries on locality, distance, and so on.

Graph processing provides the definition and querying of data that is best described as highly networked, such as a social media friends group.

Series processing allows you to analyze data that is organized into measurable time intervals such as gas meter readings so that you can perform queries on this data to identify patterns, etc.

Predictive processing supports deep analysis of data patterns in order to provide insights and probable outcomes.

Data Warehousing Foundation (DWF)

Data Warehouse Foundation (DWF) provides the extra functionality and components that are required when building a native data warehouse using only SAP HANA (not BW).

Dynamic Tiering

Dynamic Tiering provides a disk-based tier to complement the existing memory tier of the SAP HANA database. This means that you can offload less critical data to a disk-based store, thus freeing up memory for only the most important data. The disk based-data is still fully accessible to any application just as if it were in memory. However, performance will be affected. This is often acceptable for older data or less frequently used data where fast database responses are not essential.

Two- or Three-Tier Stacks with SAP HANA

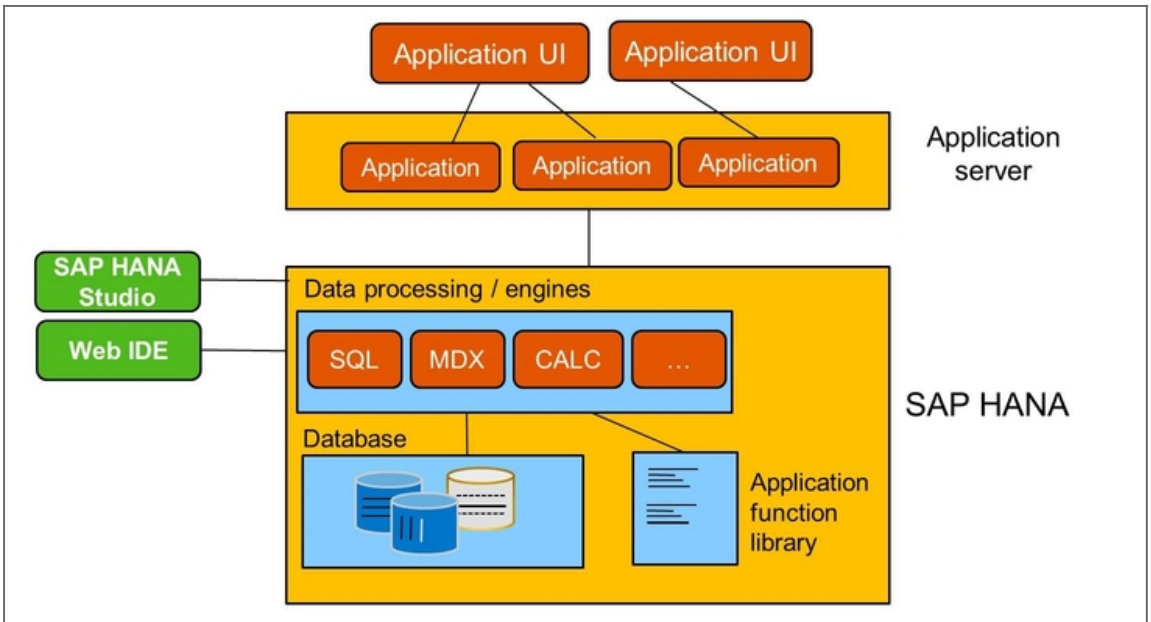


Figure 21: Typical Three Tier Landscape

SAP HANA powers many enterprise applications, including SAP S/4HANA, Business Suite, and SAP Business Warehouse (SAP BW). In the scenario shown in the figure, Typical Three Tier Landscape , SAP HANA provides the database and data processing services. On top of SAP HANA is the application server. In the case of SAP applications, this application server is usually SAP NetWeaver. SAP NetWeaver is still required to provide the business layer, the flow logic, and the connectivity and orchestration with other applications. In this case, you often find that all data in the HANA database is generated by the application (user input or

data generated from batch programs), so there may not be a need for additional external connections to SAP HANA from outside data sources.

Wherever possible, the application layer passes the processing down to SAP HANA to speed up the processing. SAP applications are always optimized to run on SAP HANA and the application code is carefully written by SAP in order to extract all the power from SAP HANA.

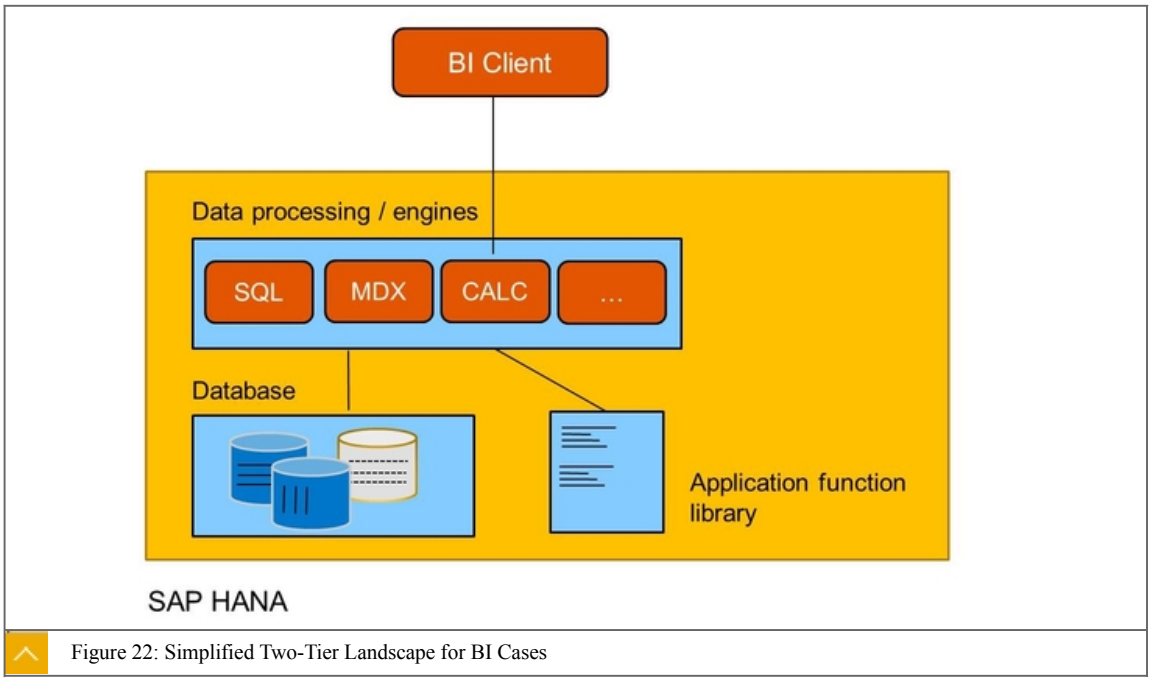


Figure 22: Simplified Two-Tier Landscape for BI Cases

For customers who use SAP HANA to provide business intelligence (BI), a simple two-tier landscape is possible. The figure, Simplified Two-Tier Landscape for BI Cases , shows that we have removed the application layer completely. The communication now takes places directly between the front-end BI tools and SAP HANA. SAP HANA connects to any SAP BI tool using JDBC, ODBC, ODBO, or BICS. The data in the SAP HANA database must be first acquired and you can use the built-in EIM components, or external data provisioning tools such as Data Services or SLT, to acquire BI data from any source.

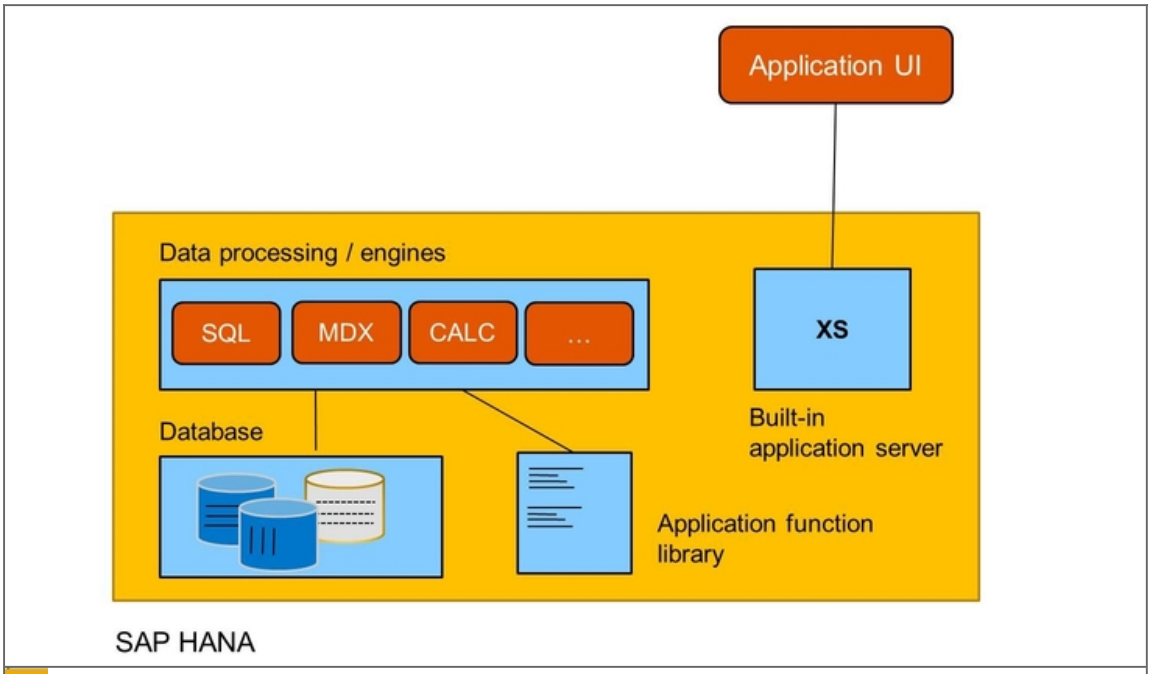


Figure 23: Simplified Two-Tier Landscape for Native SAP HANA Applications

Another implementation of a simple two-tier stack is when we develop applications using the in-built XS/XSA application server so that an external application server is not required.

For example, imagine that you want to build a simple stock-checking application for mobile device users who work in the warehouse. XS/XSA provides all the application services you need to access the required data from the SAP HANA database, process it, and pass it to the front-end interfaces.

From XS to XSA

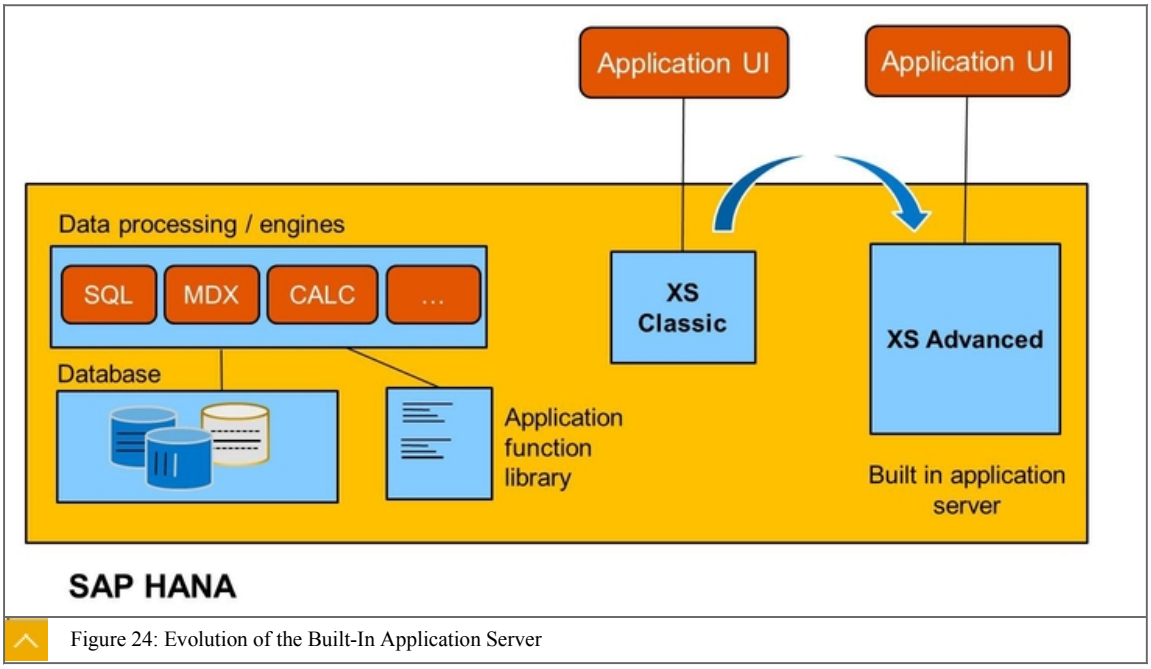


Figure 24: Evolution of the Built-In Application Server

Extended Application Services (XS) is the name of the built-in application server that was first introduced with SAP HANA SPS5. It has always been positioned as a lightweight application server. It was never meant to be used to build heavy-weight, scalable enterprise applications.

XS supports the development and run-time of Javascript and HTML-based applications. XS also includes a web server. It is fully integrated into SAP HANA and communicates directly with the SAP HANA database and, of course, processes data and application logic completely in-memory for great performance.

Since SPS11, SAP delivered a significantly more powerful version of XS. This new version is called XS Advanced (XSA). Although XS and XSA have the same overall goal, they are technically very different. This means applications developed with XS are not compatible with XSA and vice-versa.

Currently, XS and also XSA are installed side-by-side in SAP HANA and both are currently supported. But XSA completely replaces XS, and in the future only XSA will be supported. Customers are encouraged to migrate their XS application to XSA (migration tools are supplied). Also, future developments should only use XSA. In fact, as of SAP HANA 2.0 SPS02, XS became officially deprecated.

Note:
XS is often referred to as “classic” to avoid confusion with XSA.

So why did SAP implement a completely new application server framework called XSA to replace XS?

De-couple applications from infrastructure — XSA is based on the common open standard known as Cloud Foundry. This means applications developed with SAP HANA XSA can easily be deployed either on-premise or to any cloud provider (SAP Cloud Platform, Microsoft Azure, AWS, Google Cloud Platform etc.) without changing the code. This gives greater choices to customers who can decide on their own deployment options.

Supports more languages — Instead of just one application language, XSA supports multiple development languages and/or run-times including Javascript, Node.js, Java, and C++. Plus you can also create your own custom run-time to support more languages such as Python and PHP.

Mix languages in one application — An important architectural change is that XSA supports a micro-services architecture. This is a modern approach to application development where applications can be built from multiple languages. The developer chooses the most effective development language for each part of the application and the run-times are combined to form a complete application. It also means it is easy to integrate other external services into your applications. And finally, it is now possible to configure each part of the application to consume more or less resources as needed. This is known as elastic computing.

Improve source code management — To enable the use of common industry standards for source code sharing and version control. XSA is fully integrated with the very popular Git/ GitHub/Gerrit source code management solutions. SAP HANA no longer stores and manages the source code.

LESSON SUMMARY
You should now be able to:

- Outline components of SAP HANA

Unit 2
Lesson 2

Working with SAP HANA interfaces

LESSON OVERVIEW
This lesson covers the interfaces used by administrators and developers.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Work with SAP HANA interfaces

SAP HANA Studio

Although there are various interfaces used to interact with SAP HANA, SAP HANA Studio was the original interface and is still used today, especially for customers running SAP HANA 1.0. SAP HANA Studio can also be used with SAP HANA 2.0 when working in the XS classic development mode. However, SAP HANA Studio cannot be used to develop XSA applications as it does not work with the new HDI infrastructure. SAP HANA Studio is a powerful multi-purpose interface used by several user roles including developers, administrators, and modelers. SAP HANA Studio is a Java application that runs on Windows, Apple MacOS, and Linux. It is based on the well-known, open-source Eclipse and it includes many add-ins provided by SAP to support SAP HANA.

Note:
Although SAP HANA Studio is relevant today, especially for SAP HANA 1.0, it is important to remember that since SAP HANA 2.0, the new interface, SAP Web IDE for SAP HANA, is the focus for new functionality developed by SAP. It is taking over the role of SAP HANA Studio, especially for developers and modelers. The administration tasks performed by Studio are also now moving to SAP HANA Cockpit.

Introducing Connections

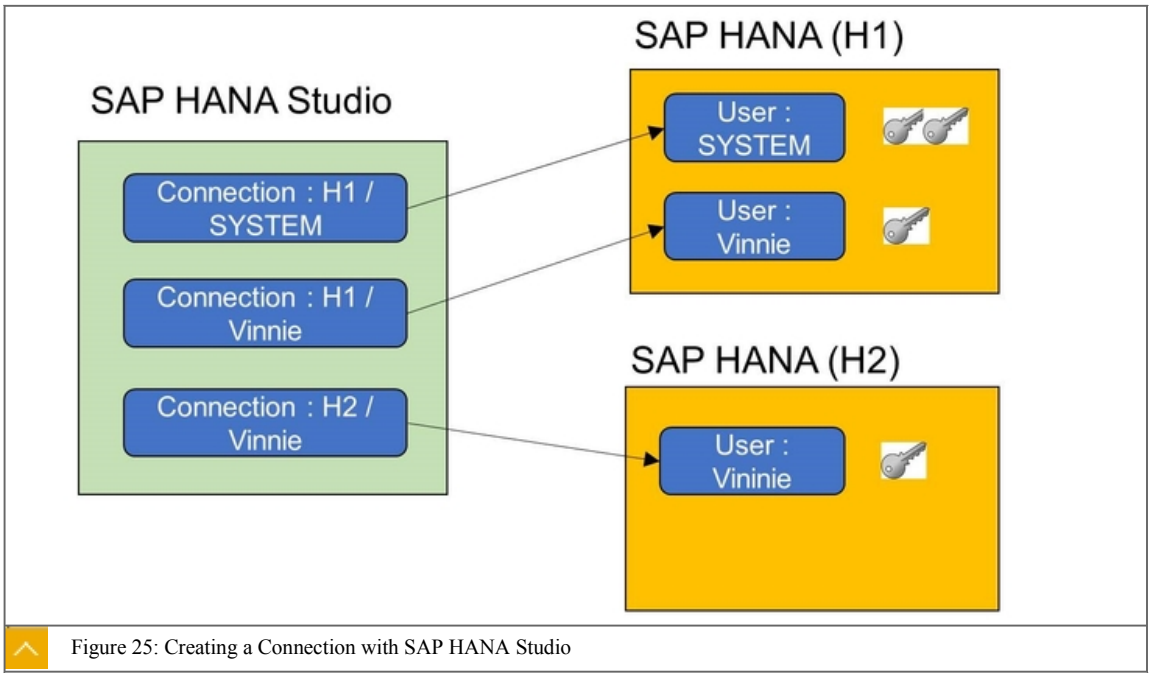


Figure 25: Creating a Connection with SAP HANA Studio

To get started with SAP HANA Studio, you must first create a **connection** to the SAP HANA target system that you want to work with.

A connection is a link between SAP HANA Studio and the SAP HANA platform. After opening SAP HANA Studio, you can create as many connections as you wish to point to different SAP HANA Platforms. This means that you can open the SAP HANA Studio once and then navigate between SAP HANA platforms easily, and all on one screen.

You can optionally give each connection a description so it is easy to identify the purpose of each system when the list of connections becomes long. For example, to identify TEST, DEV, PROD, QA, and so on.

When defining the connection you will be asked to specify some important information, as follows:

- The host and instance (identifies the exact HANA target system).
- A valid SAP HANA database user and password that has been given sufficient privileges on the target SAP HANA system for you to access what you need.

When you log off and then back on to the SAP HANA Studio where connections already exist, you will need to provide the passwords for each connection again in order to use them.

It is possible to export the list of connections to a file, so these can then be imported by others. Doing this would mean that they do not have to manually define the connections. Of course the user credentials are not saved and need to be re-entered.

You can also use the exported list of connections and share them as a central store. Each user creates a link to this central store, and thus does not need to create their own connections or import connections. This means that all connection information is managed centrally, so any changes are made in just one place.

Introducing Perspectives and Views

When you start the SAP HANA Studio for the first time, you are presented with the Welcome page. From here you select a **perspective** . Perspectives are simply predefined layouts that contain several panes, or **views** as they are referred to in SAP HANA Studio. One or more perspectives address the needs of a particular SAP HANA user role. For example, a System Administrator uses the SAP HANA Administration Console perspective; a BI person might use the SAP HANA Modeler perspective. You can switch between the perspectives as you wish to access the features you need. You can even create a custom perspective so you can design your own layout to suit your needs.

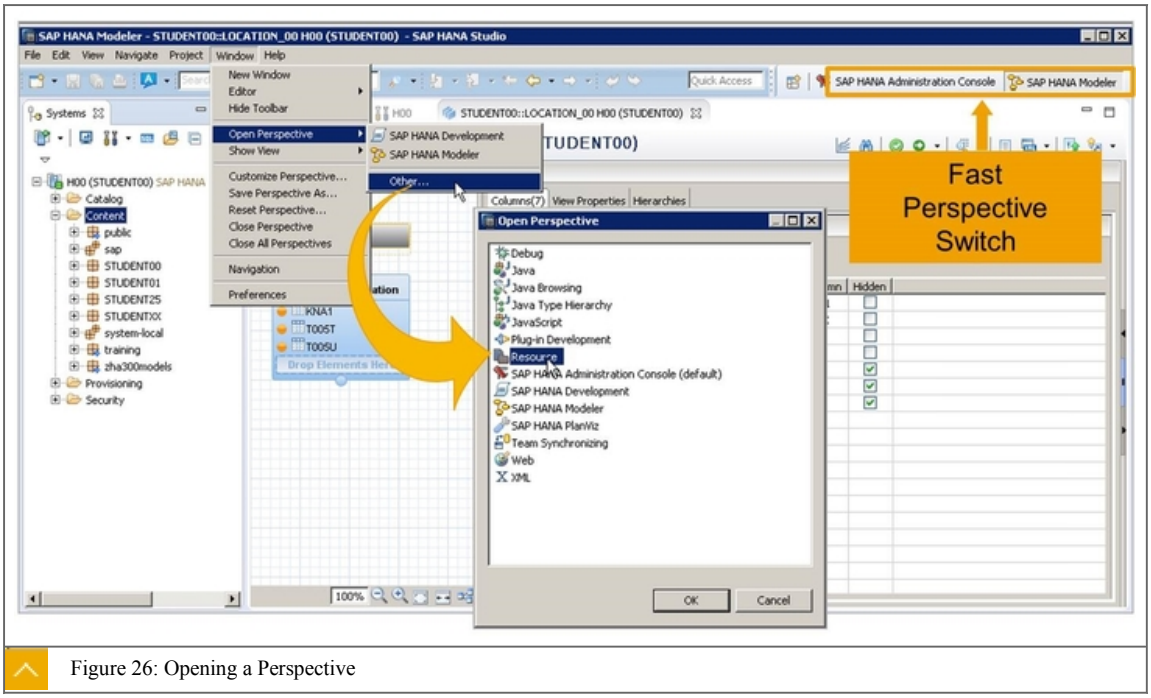


Figure 26: Opening a Perspective

The following are examples of some perspectives that are available in the SAP HANA Studio:

- SAP HANA Modeler

  The SAP HANA Modeler perspective is used by Data Modelers to create Information Models, as a combination of attributes, dimensions, and measures, included in different types of modeling views.

- Administration Console

  The Administration Console perspective is used by SAP HANA Administrators to administrate and monitor the whole SAP HANA system.

- Development

  The Development perspective is used by application developers (coders) to build native SAP HANA applications.

A **view** is simply an individual pane, within a perspective, that provides specific information, such as a Where Used list or Properties . Each view can be moved around via drag and drop. You can also customize the perspectives by adding or removing views. Views can appear in multiple perspectives. For example, the System view is used in most perspectives as it presents a hierarchical list of objects in each SAP HANA system that is useful to every role.

Perspectives are Based on Views

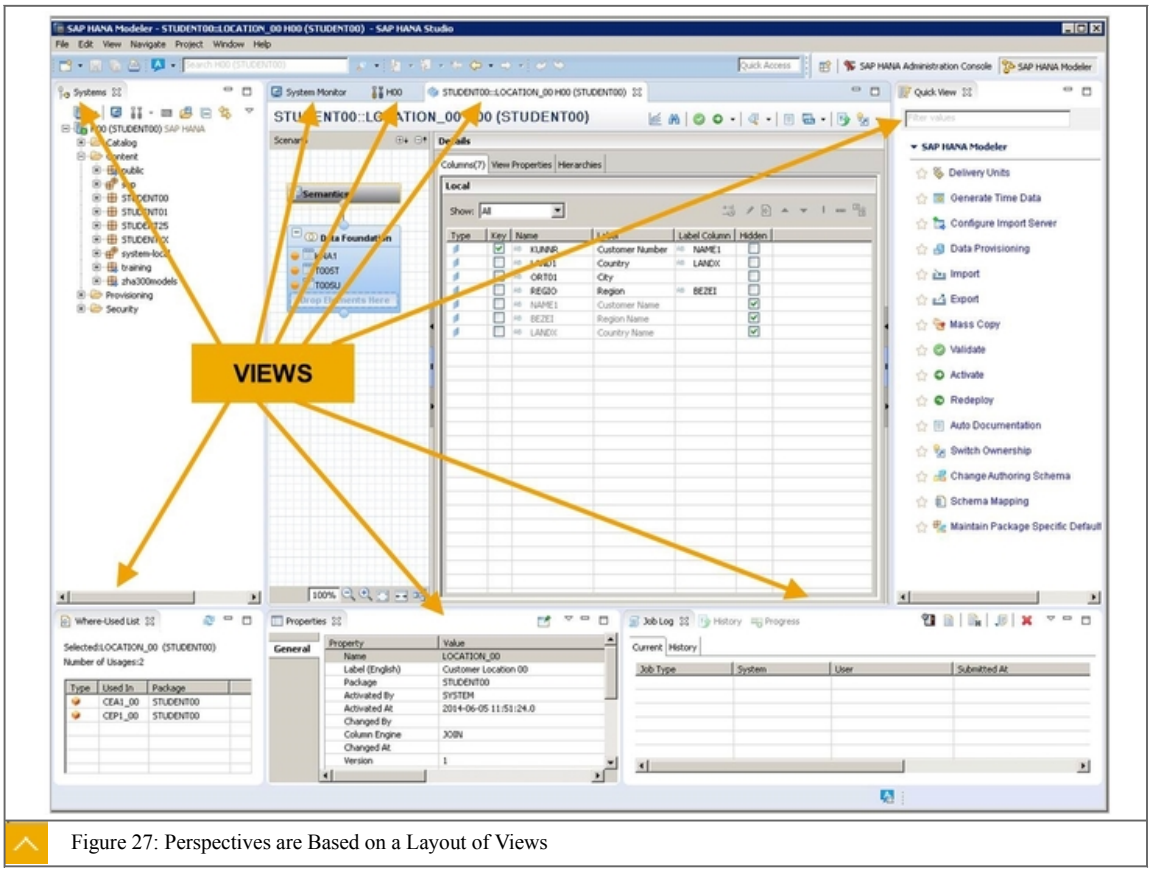


Figure 27: Perspectives are Based on a Layout of Views

There are many other perspectives in the SAP HANA Studio.

To open a perspective, choose Windows → Open Perspective . Choose a perspective from the list or choose Other... .

It is possible to have several perspective open at the same time, and to switch from one perspective to another. To do so, in the top right of the screen, choose the perspective button you want to switch to. There, you will also find a button that is a shortcut to the perspective selector.

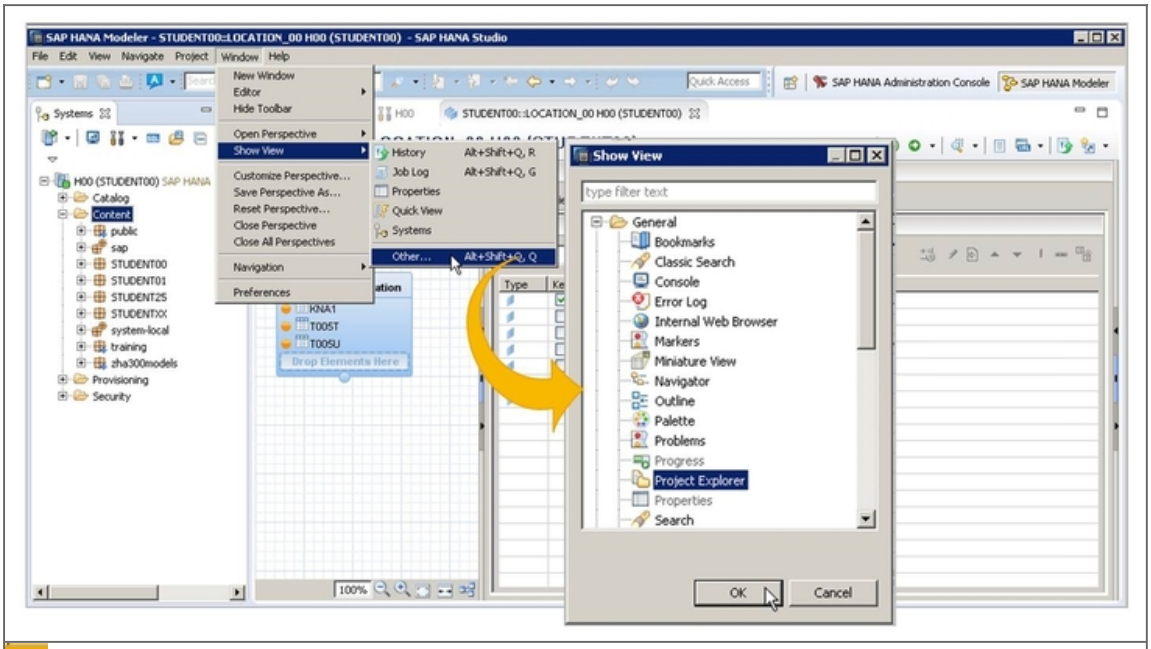


Figure 28: Adding a View to a Perspective

To add a view to a perspective, choose Windows → Show View . Choose a view from the list, or choose Other... . The view you have chosen will be placed on your screen in a default location and with a default size but you can drag it to a new location as required and resize.

Any perspective can be reset to its default layout to restore the default views in their original positions and sizes. To do so, choose Window → Reset Perspective... .

The Systems View

Some views are more important than others. The Systems view is one of the most important so let's give it some focus here.

The Systems view is used to navigate the various content of SAP HANA. This includes the run–time contents of the SAP HANA database and also the design-time modeling content. You can also list the remote sources and their contents. You can list and manage users and roles.

In the Systems view you will find the connections. You can expand each connection and explore the contents. For each connection, the content is organized as follows:

- Catalog
- Content
- Provisioning
- Security

Catalog

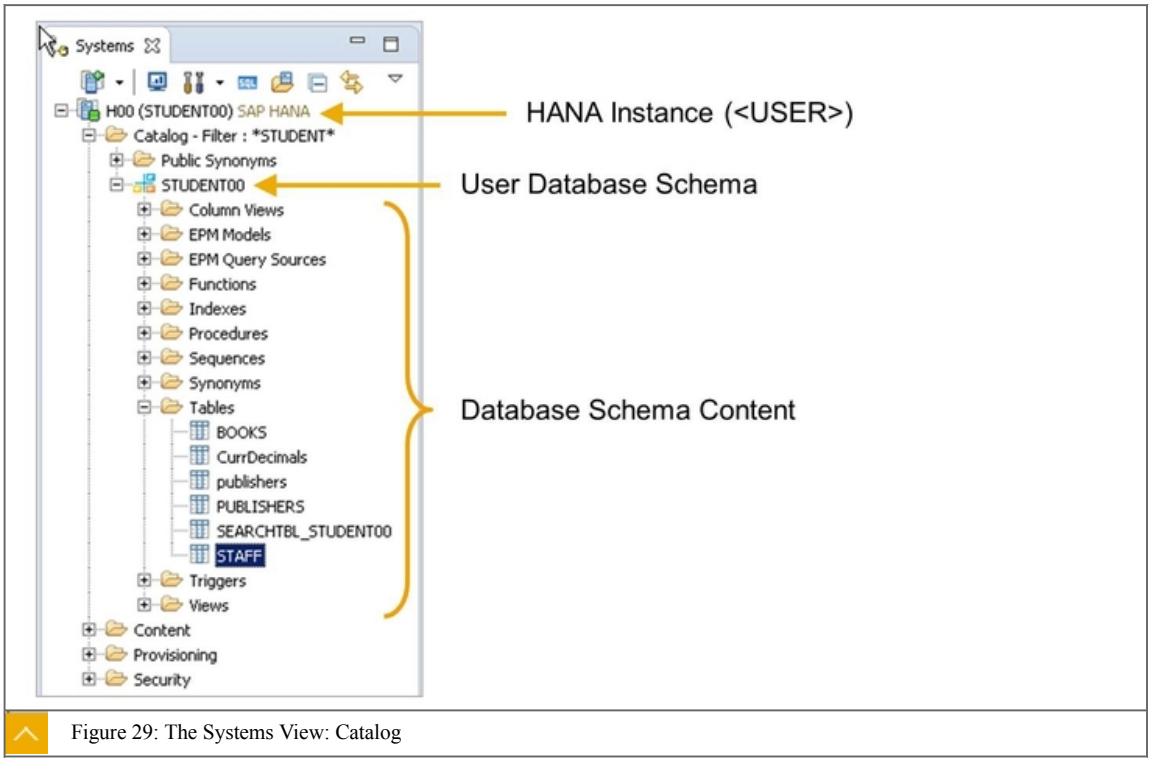


Figure 29: The Systems View: Catalog

The catalog contains all database objects such as tables, views, functions and indexes. All these objects are organized into schemas. Schemas are used to categorize database content according to customer-defined groupings that have a particular meaning for applications. Schemas are also used to define access rights to the database objects that they contain.

Content

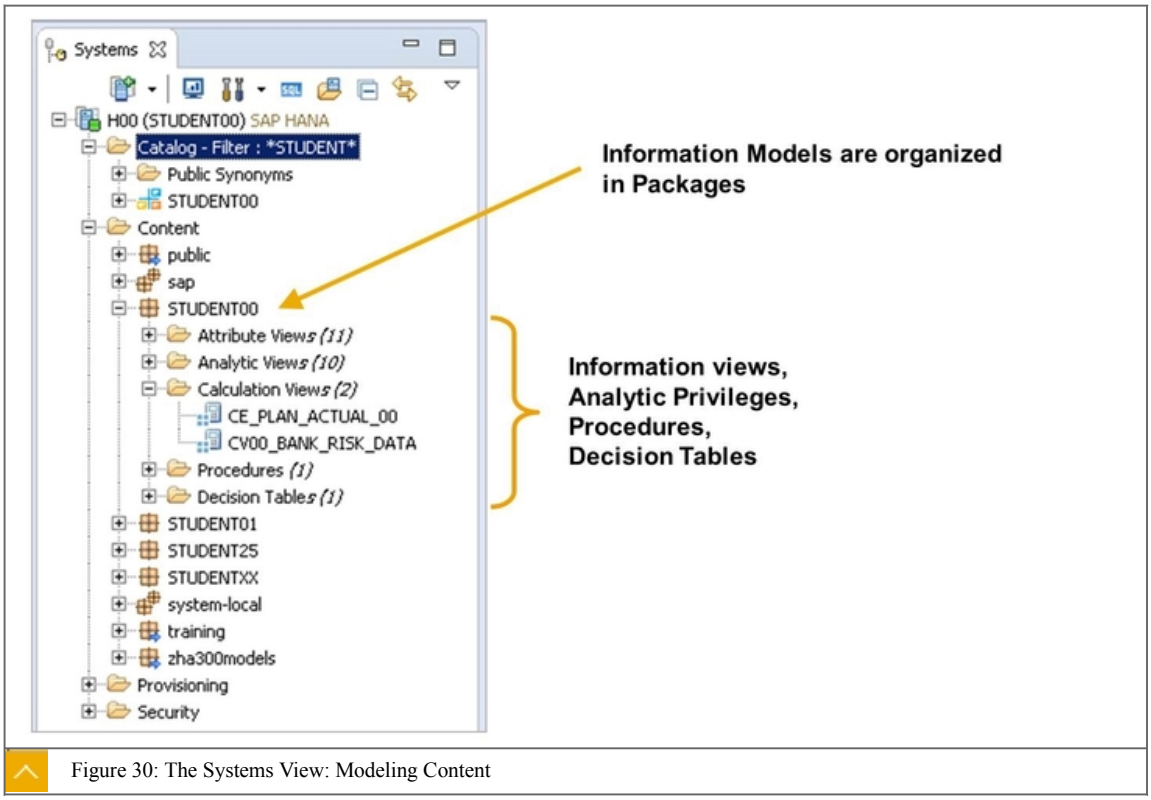


Figure 30: The Systems View: Modeling Content

The Content folder is where you store all the HANA-specific design-time data modeling objects such as calculation views and procedures. The modeling objects are organized in packages. Packages are used to define privileges to the developers to allows access only to the packages they are supposed to use. Packages are also used to transport the related objects to different SAP HANA systems.

Provisioning

The Provisioning folder is related to Smart Data Access (SDA). SDA is a data provisioning approach in which you can combine data from remote sources (Hadoop, SAP ASE, SAP IQ) with data of your SAP HANA physical tables, by exposing them as virtual tables.

Security

In the Security folder, the System Administrators define users and roles.

What you will not find in the Systems view are application development artifacts such as Javascript files, HTML code, core data services, and graphics used in application GUIs (such as buttons, etc.). Development artifacts are found under the Project view which is part of the Development perspective.

The Administration View

The SAP HANA Administration View is another important view so lets focus a little on this.

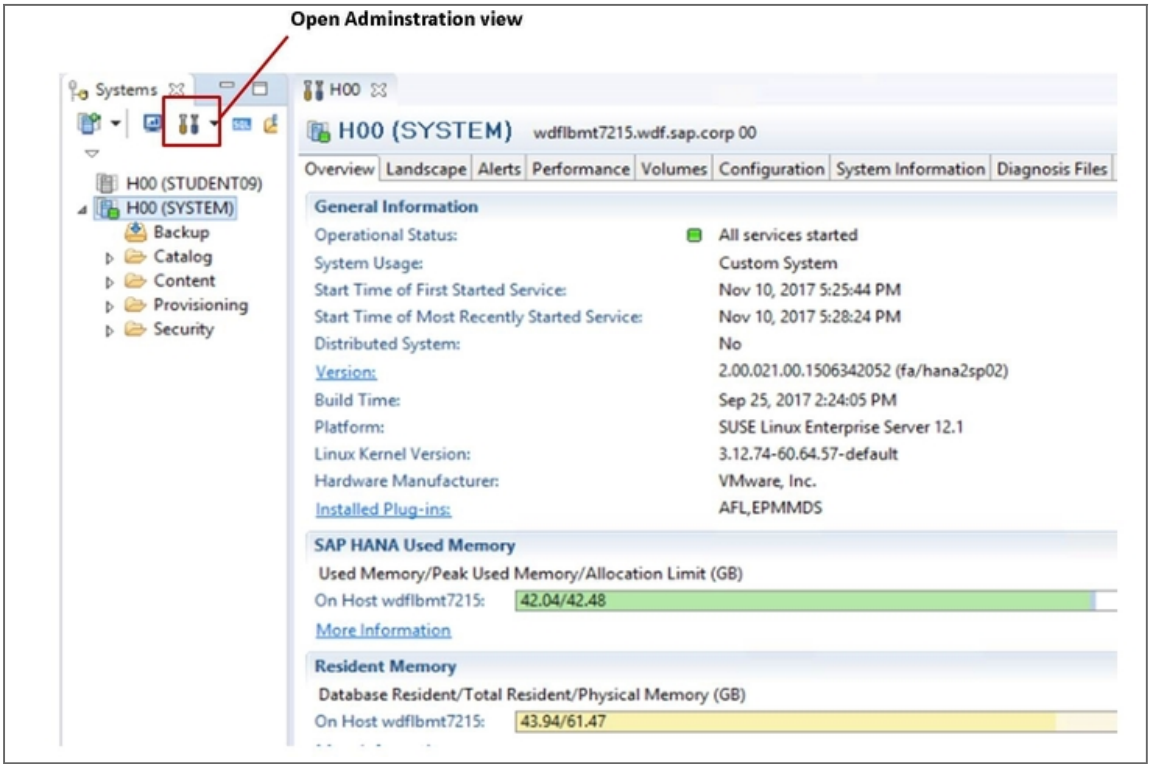


Figure 31: Administration View of HANA Studio

The Administration view is part of the Administration perspective. This is the main view used to administrate SAP HANA systems. In this view, you can do the following tasks:

- Start and stop a system
- Configure a system
- Monitor a system

- Backup and restore a system
- Perform a problem analysis

To open the Administration view, you can either double-click the required SAP HANA system listed in the System Monitor view or use the button in the toolbar of the Systems view.

Quick View

The Quick View contains a number of popular functions and basically puts them into a mini–menu for your convenience. When you want to perform an action from the Quick View, you must first select the SAP HANA system on which the action will be executed.

To access the Quick View you can:

- Choose Help → Quick View.
- Reset the Modeler perspective.

Note:
The Quick View only displays within the Modeler perspective.

In summary, with the different perspectives and views of SAP HANA Studio, you can easily reach all the tools to develop complete applications from one interface. This ranges from the creation of database tables, to developing models to expose the table data in meaningful views, to developing applications that consume the views, and finally to monitoring performance of the applications.

Unit 2
Exercise 2

Get Started with SAP HANA Studio

The purpose of this exercise is to set up and become familiar with the SAP HANA Studio user interface. You will perform the following tasks:

- Start SAP HANA Studio.
- Add an SAP HANA connection to the Systems view.
- Define SAP HANA Studio preferences.
- Work with Views and Perspectives .

Business Example

You are working on a customer project where SAP HANA is installed. You need to become familiar with SAP HANA Studio and learn how to connect to the SAP HANA system and browse the database catalog objects as well as the repository.

1. Launch the SAP HANA Studio.

   If you are prompted to choose a folder to store settings, use the default location and choose Submit .

   If you are prompted to choose a workspace folder, leave the defaults unchanged and choose OK.

   If you are asked to create a password hint (in case you forget your password), choose No.

2. From the Welcome page, open the SAP HANA Administration Console perspective.

3. Connect your SAP HANA Studio to the SAP HANA training system using the details shown in the table, Connection Information .

   Table 2: Connection Information

Field	Value
Host Name	**wdflbmt7215.wdf.sap.corp**
Instance Number	**00**
Database Mode	Multiple containers
Database name	**H00** (Tenant Database)
Description	**My SAP HANA Training system**

4. Enter the credentials shown in the table, User Credentials .

Table 3: User Credentials

Field	Value
User	STUDENT## (don't forget to swap ## for your 2 digit number)
Password	**`Training1`**

5. Set your SAP HANA Studio preferences using the values shown in the table, Preferences .

Table 4: Preferences

Path and Parameters	Value
General → Network Connections → Active Provider	**`Direct`**
General → Web Browser → Use External Web Browser	**`Internet Explorer`**

6. You are currently working in the SAP HANA Administrator perspective. Switch to the SAP HANA Modeler perspective.
7. Switch back to the SAP HANA Administration Console perspective for the next tasks.
8. Explore the Systems view by first increasing the width of the view to occupy half the screen and then expand the following nodes.
   - Catalog
   - Content
   - Provisioning
   - Security

Note:
You can temporarily expand any view to full screen by simply double-clicking on the tab. You can restore the view to its original size by repeating this action. This can be very helpful as many of the views contain a lot of information that is easier to read when you fill the whole screen.

9. Open the Administration view for your SAP HANA system.

Unit 2
Solution 2

Get Started with SAP HANA Studio

The purpose of this exercise is to set up and become familiar with the SAP HANA Studio user interface. You will perform the following tasks:

- Start SAP HANA Studio.
- Add an SAP HANA connection to the Systems view.
- Define SAP HANA Studio preferences.
- Work with Views and Perspectives .

Business Example

You are working on a customer project where SAP HANA is installed. You need to become familiar with SAP HANA Studio and learn how to connect to the SAP HANA system and browse the database catalog objects as well as the repository.

1. Launch the SAP HANA Studio.

   If you are prompted to choose a folder to store settings, use the default location and choose Submit .

   If you are prompted to choose a workspace folder, leave the defaults unchanged and choose OK.

   If you are asked to create a password hint (in case you forget your password), choose No.

   a) Locate and launch SAP HANA Studio with the default settings.

2. From the Welcome page, open the SAP HANA Administration Console perspective.

   a) Choose Open SAP HANA Administration Console .

3. Connect your SAP HANA Studio to the SAP HANA training system using the details shown in the table, Connection Information .

   Table 2: Connection Information

Field	Value
Host Name	`wdflbmt7215.wdf.sap.corp`
Instance Number	`00`
Database Mode	Multiple containers
Database name	`H00` (Tenant Database)
Description	`My SAP HANA Training system`

   a) In the Systems view, right-click the blank area and choose Add System .

b) Enter the details from the table, Connection Information, in the appropriate fields.

c) Choose Next .

4. Enter the credentials shown in the table, User Credentials .

Table 3: User Credentials

Field	Value
User	STUDENT## (don't forget to swap ## for your 2 digit number)
Password	**Training1**

a) Enter your user name and password shown in the table, User Credentials .

b) Choose Finish .

5. Set your SAP HANA Studio preferences using the values shown in the table, Preferences .

Table 4: Preferences

Path and Parameters	Value
General → Network Connections → Active Provider	**Direct**
General → Web Browser → Use External Web Browser	**Internet Explorer**

a) Choose Window → Preferences and follow the path provided in the table, Preferences , to set the values.

b) When finished, select OK.

6. You are currently working in the SAP HANA Administrator perspective. Switch to the SAP HANA Modeler perspective.

a) To display all available perspectives, choose Window → Perspective → Open Perspective → Other .

b) Choose the SAP HANA Modeler perspective.

Note that the layout changes slightly to provide features that a modeler uses.

7. Switch back to the SAP HANA Administration Console perspective for the next tasks.

a) Using the perspective switch buttons located in the upper-right corner of the screen, choose SAP HANA Administration Console .

8. Explore the Systems view by first increasing the width of the view to occupy half the screen and then expand the following nodes.

- Catalog
- Content
- Provisioning
- Security

Note:
You can temporarily expand any view to full screen by simply double-clicking on the tab. You can restore the view to its original size by repeating this action. This can be very helpful as many of the views contain a lot of information that is easier to read when you fill the whole screen.

a) Drag the right border of the Systems view toward the center of the screen to increase the size.

b) Choose the arrow to the left of each of the four nodes to expand the details and explore the tree.

9. Open the Administration view for your SAP HANA system.

a) In the Systems view, choose the H00 (STUDENT##) system.

b) Choose the Administration icon (looks like two screwdrivers) in the tool bar at the top of the Systems view

c) Explore the system information.

Working with Web-based Development Workbench

SAP Web based Development Workbench

In addition to SAP HANA Studio, for XS developments there is the SAP HANA Web-based Development Workbench.

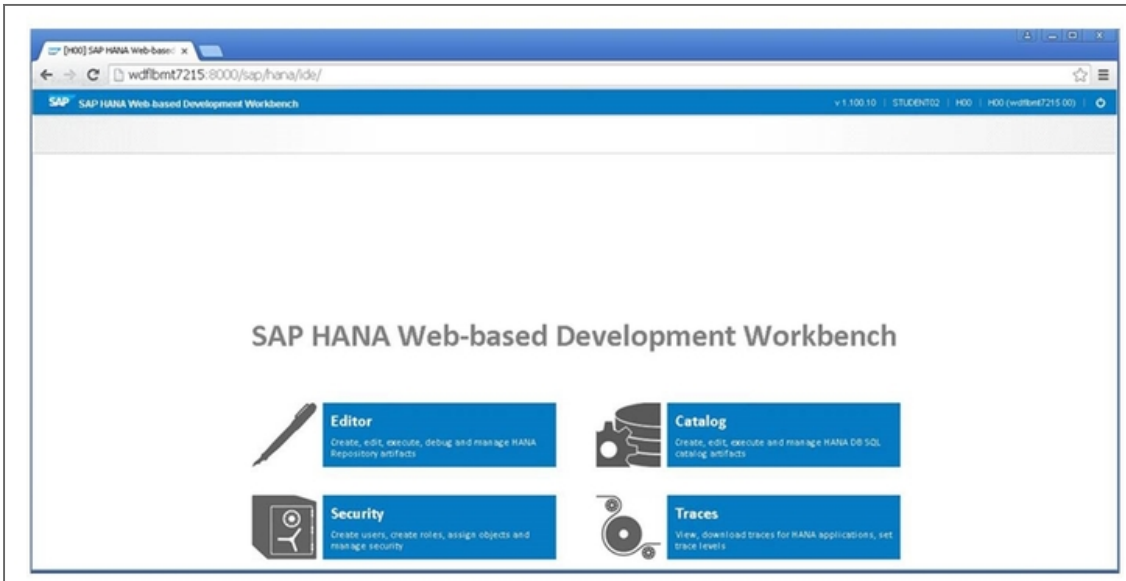


Figure 32: SAP Web Based Development Workbench

This lightweight web-based interface was aimed at developers and coders as an alternative to using the heavier SAP HANA Studio. The SAP HANA Web Development Workbench focused only on the features needed by application developers. You cannot use this interface with XSA developments.

Note:
Be careful not to confuse SAP Web based Development Workbench with the Web IDE for SAP HANA.

In order to get started with the SAP Web based Development Workbench, open a web browser and enter the following URL:

`http://<hostname>:80<instance>/sap/hana/xs/ide`

Substitute the two placeholders with your system details.

Enter your SAP HANA user and password and choose LOG ON.

Working with SAP Web IDE for SAP HANA

SAP Web IDE for SAP HANA is a browser-based, integrated development environment (IDE). It is used for the development of SAP HANA-based applications comprised of web-based or mobile UIs, business logic, and extensive SAP HANA data models. SAP Web IDE works in conjunction with the SAP HANA deployment infrastructure (HDI) and the XS Advanced runtime platform. It does not work with XS Classic development.

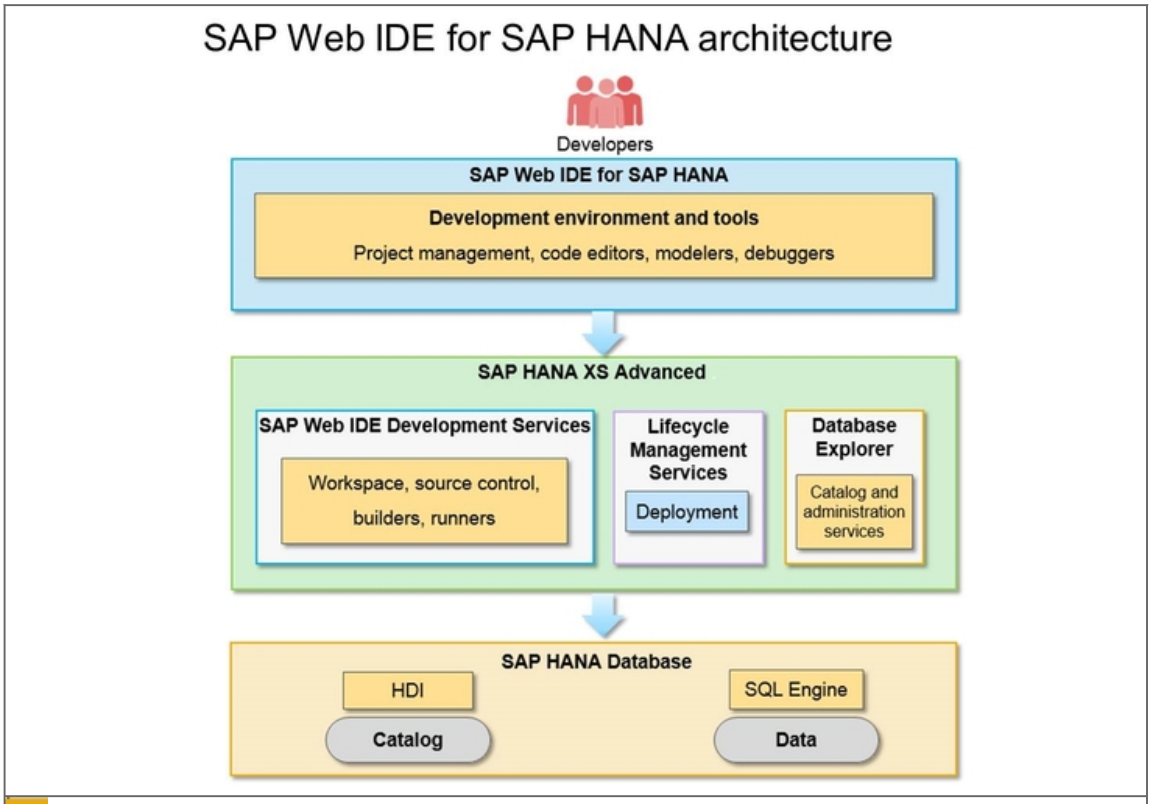


Figure 33: SAP Web IDE for SAP HANA Architecture

Note:
It is important to remember that SAP Web IDE for SAP HANA is not the same as SAP Web IDE. SAP Web IDE is a very similar client tool, but it is used for the development of Fiori-based applications and SAPUI5 applications. SAP Web IDE for SAP HANA is used to build XSA applications on SAP HANA. Be careful when referring to documentation as these products are easily confused.

SAP Web IDE for SAP HANA and the database explorer enhance the development experience by providing the following features:

- **Integrated Workspace and Project Management**
  - The latest tool versions are always available without additional installation.
  - Full workspace management on the server
  - Integrated Git-based version control
  - A dedicated template and wizards for multi-target application projects
- **Development Tools**

  For SAP HANA database artifact development:
  - Graphical modelers for complex artifacts, such as data models or calculation views
  - An SQL console and an MDX console in the database explorer

- Integrated browsing of the database catalog in the database explorer
- Syntax highlighting and content assistance for selected artifacts
- Integrated performance analysis tools for SQL script and calculation views

- **Build, Run, and Deploy**
  - Incremental build support
  - Structured console output and logging
  - Automatic creation of development sandboxes on SAP HANA (HDI containers)
  - Automatic provisioning of HDI container services
  - Generation of deployment archives

To launch the Web IDE for SAP HANA, you need to obtain the URL from the administrator.
This will include the host name where HANA is installed, which is unique for each installation.
After launching the URL, the screen shown in the figure, Launching Web IDE for SAP HANA , will be displayed.

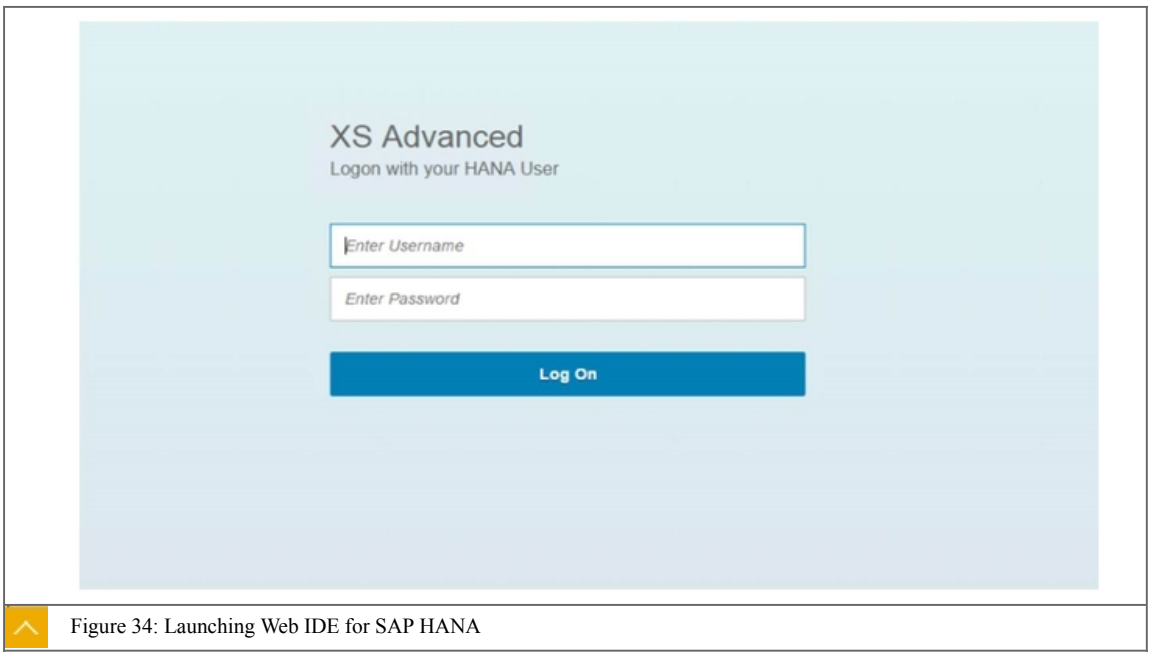


Figure 34: Launching Web IDE for SAP HANA

Once you have logged on, you can then decide which view you want to work in.

Switching Views in Web IDE

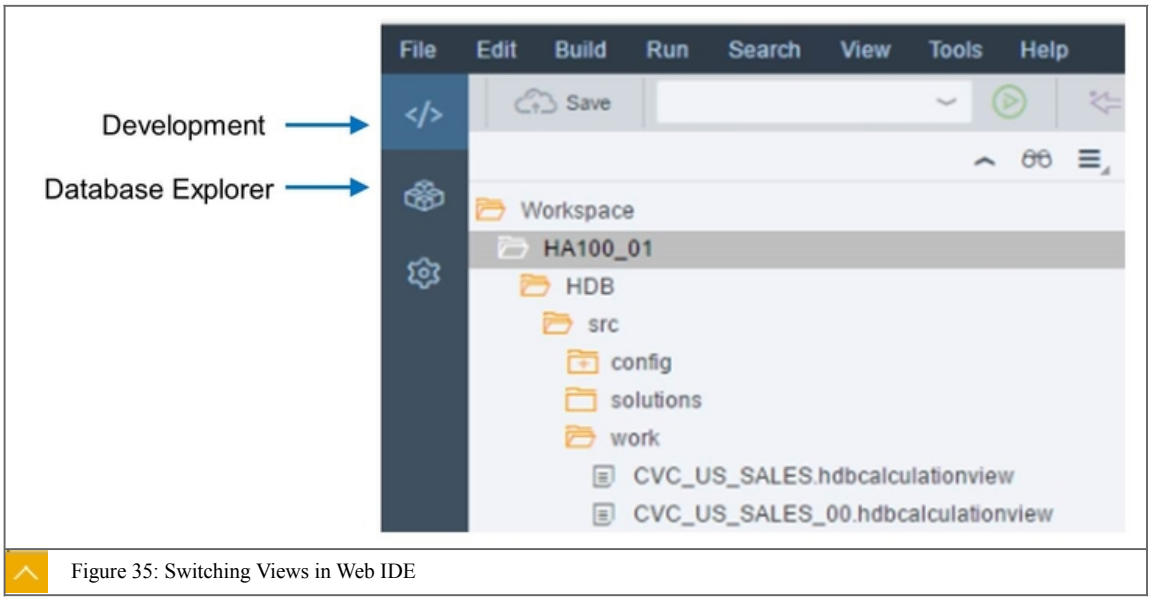


Figure 35: Switching Views in Web IDE

There are two main views in Web IDE for SAP HANA. These are the Development view and the Database Explorer view.

Development View of Web IDE for SAP HANA

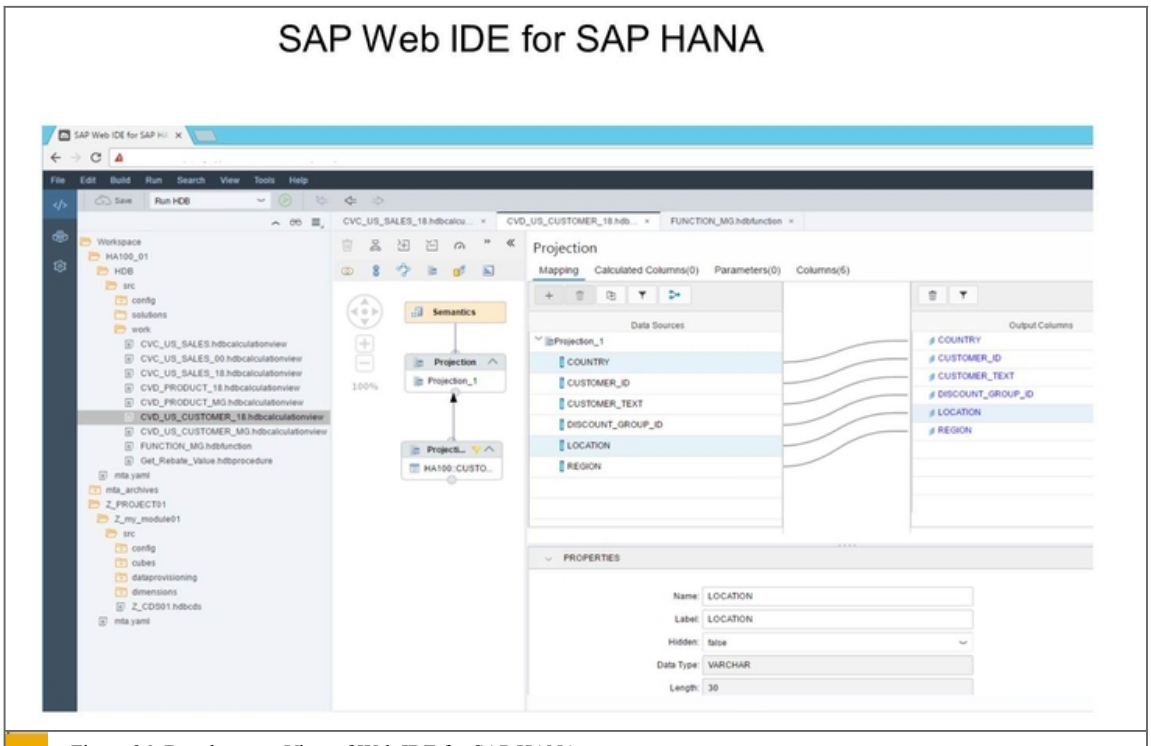


Figure 36: Development View of Web IDE for SAP HANA

The Development view is used by application developers to write code. It is also used by modelers to develop data models. In comparison to SAP HANA Studio, it is like having the Modeling perspective and the Development perspective combined.

The Development view has some convenient built-in features for developers, such as code completion and debuggers.

> Note:
> Remember Web IDE for SAP HANA is used only with the newer XSA and HDI based development. If you are working on classic HANA artifacts in XS, then use Studio or the Web-based Development Workbench.

Creating a New Development Module

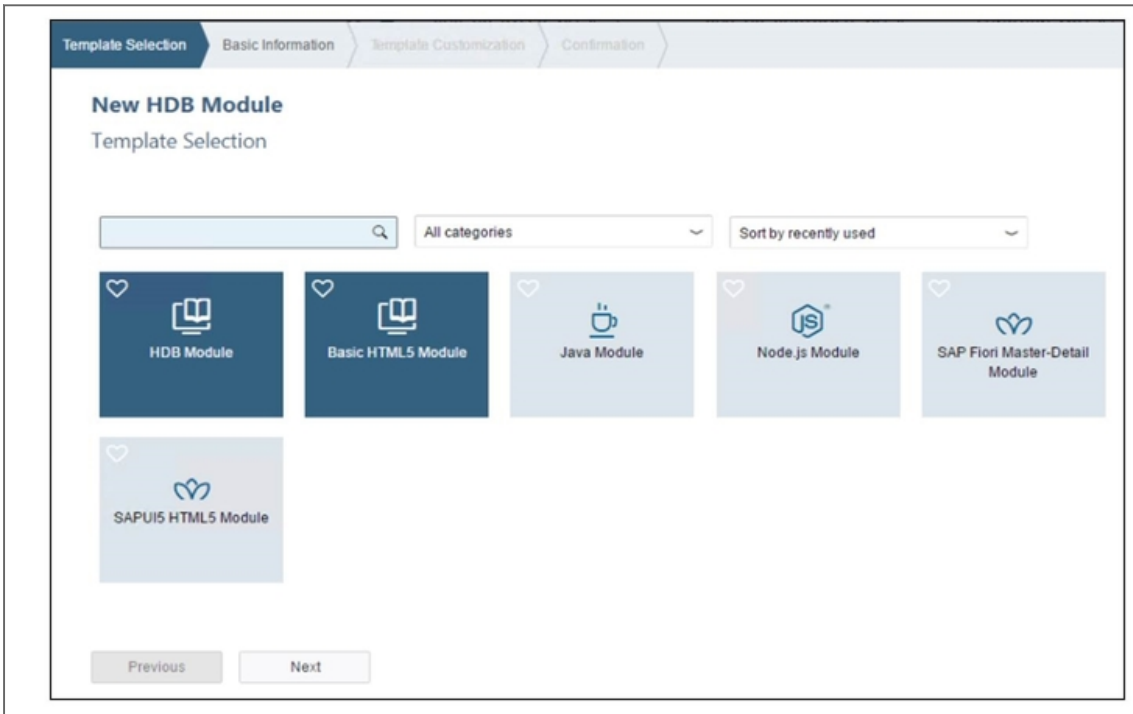


Figure 37: Creating a New Development Module

To get started with modeling or application development, you first need to create a development module. This provides the templates and structure that you will need.

Once your module is built, you can add additional folders to provide better organization of your content.

To create the various artifacts and source files, you right-click a folder and choose the type of file in the context menu.

As an example, if you were a modeler you would be able to create the following:

- Procedure
- Function
- Calculation View
- Flowgraph
- Analytic Privilege
- CDS Artifact

- Text Analysis dictionaries and rule sets

Web IDE development is completely file based, which means that each source artifact is a simple file (identified with an extension) that is easy to export and import within and also across SAP HANA.

Each file you create in Web IDE has its own extension, such as ".hdbfunction" or ".hdbcalculationview", which makes it easy to identify its type. This is also how Web IDE knows which editor to open for each type of artifact.

Database Setup in Web IDE

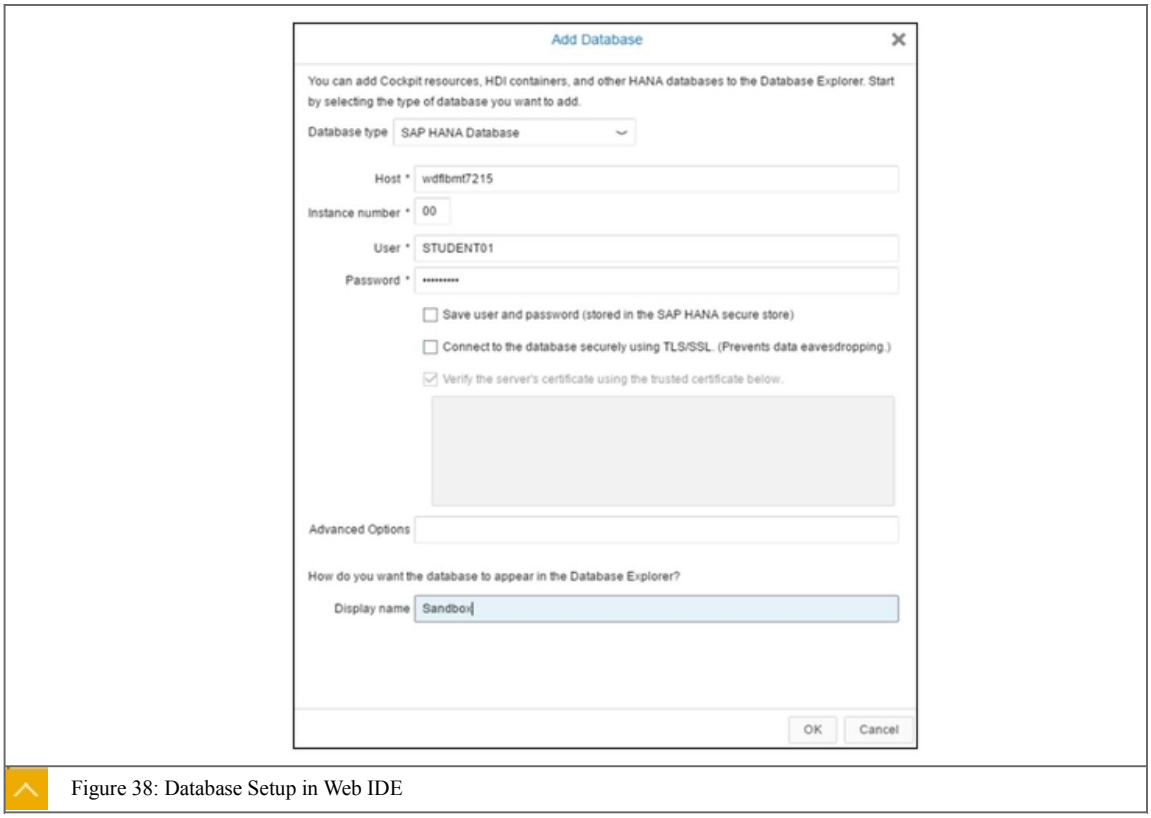


Figure 38: Database Setup in Web IDE

The Database Explorer view is used by anyone who needs to view the runtime database objects such as tables, views, and functions.

There are two ways to view the database in the Database Explorer; either using the HDI container infrastructure or the classic schema-based catalog. You can switch between the two views. To get started with the Database Explorer , you need to first add a database entry to the navigation pane on the top-left side of the screen. Whilst doing this you will be asked if it is based on the classic schema or new HDI container.

After adding the database entry, you can then browse the database objects. There are tools to view the metadata and also the data of tables.

The Database Explorer also includes an SQL and MDX editor so you can execute statements directly.

Exploring the Database with Web IDE

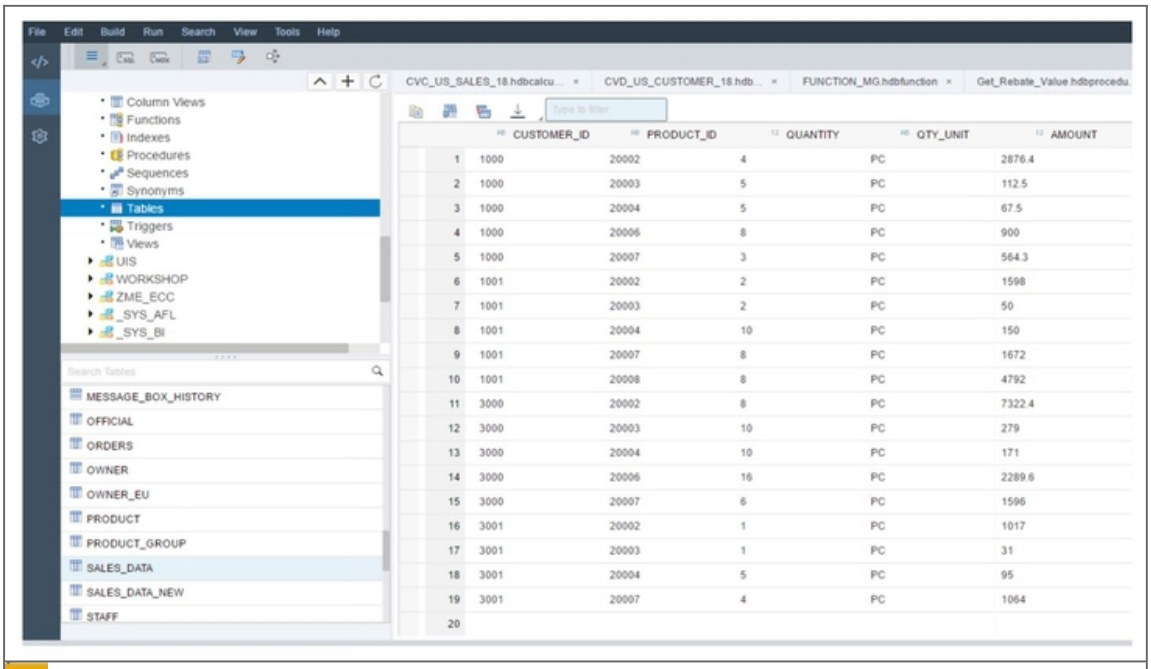


Figure 39: Exploring the Database with Web IDE

Let's briefly describe a **HDI container**.

A HDI container (or sometimes called a schema-container or just container) is a logical 'box' that sits above the physical database schema. The basic idea is to allow development to be de-coupled from the physical database storage components. Or, put simply, the developers never need to be concerned about the physical database deployment. Actually, when a container is first built, a physical schema is automatically generated and bound this container. So developers simply refer to the container and they must never specify the physical schema. (This is called schema-less development). The HDI container is a key component in the new HANA deployment infrastructure (HDI). HDI is only used with XSA, not XS.

When created for the first time, a container generates its own physical schema with a long technical name. You can create database objects directly in your container and they will be physically placed in a generated physical schema.

One of the key benefits of getting away from tying development to physical schemas and instead using containers, is that different versions of the database can co-exist in the same SAP HANA system. For example, you could upgrade an application (but keep the older version running), and redeploy the new version along with its improved database objects and a different schema will be used for the newer applications database objects. The older application will continue to run alongside using its own database schema.

Note:
With XS development, all database objects from all applications were deployed to one giant fixed system schema (called _SYS_BIC). This meant you could never have multiple versions of the database objects supporting different versions of applications.

Unit 2
Exercise 3

Get Started with Web IDE for SAP HANA

The purpose of this exercise is to launch, and become familiar with SAP Web IDE for SAP HANA. You will perform the following tasks:

- Launch Web IDE for SAP HANA.
- Navigate between the Development and Database Explorer views.
- Import a project to your workspace.
- Add a database to the Database Explorer.

Business Example

You are joining a customer project where SAP HANA XSA developments are in progress, so you need to become familiar with SAP Web IDE for SAP HANA.

Task 1: Log on to Web IDE for SAP HANA

1. Launch the Web IDE for SAP HANA.
2. Log on using the details shown in the table, Logon Details . Allow Google Chrome to save your password so you don't have to re-enter this next time you need to logon.

   Table 5: Logon Details

Field	Value
User	**`STUDENT##`**
Password	**`Training1`**

Task 2: Import a ready-made project and build the run-time database objects

Setting up a brand new Web IDE project requires a number of configuration steps. To keep things simple in this class, and to get started quickly, we will import a ready-made project that already has many of the settings in place. It also includes all the exercise solutions for this class.

1. Import an existing XSA project into your workspace.

   The project archive, HA100.zip , is located in your HA100 folder, and must be imported to your workspace with the project name **`HA100_##`** (where ## is your group number).
2. Assign your project to the DEV space.
3. Build the HDB module of your project so that the database container can be generated.

Caution:
Make sure you execute the build at the HDB module level. If you build the project, the HDB module will not be built.

4. Expand the src folder in your imported project until you reach the solutions sub–folder, where various development artifacts already exist.
5. Open the file, CVC_US_SALES_00.hdbcalculationview , in the graphical calculation view editor , and study the definition of the calculation view.

Note:
You can temporarily expand any view to full screen by simply double-clicking on the tab. You can restore the view to its original size by repeating this action. This can be very helpful as many of the views contain a lot of information that is easier to read when you fill the whole screen.

6. Close the calculation view CVC_US_SALES_00.hdbcalculationview

Unit 2
Solution 3

Get Started with Web IDE for SAP HANA

The purpose of this exercise is to launch, and become familiar with SAP Web IDE for SAP HANA. You will perform the following tasks:

- Launch Web IDE for SAP HANA.
- Navigate between the Development and Database Explorer views.
- Import a project to your workspace.
- Add a database to the Database Explorer.

Business Example

You are joining a customer project where SAP HANA XSA developments are in progress, so you need to become familiar with SAP Web IDE for SAP HANA.

Task 1: Log on to Web IDE for SAP HANA

1. Launch the Web IDE for SAP HANA.
   a) To launch the Web IDE logon, double-click the shortcut Web IDE for SAP HANA in the folder Favorite → HA100 → URLs.
2. Log on using the details shown in the table, Logon Details . Allow Google Chrome to save your password so you don't have to re-enter this next time you need to logon.

   Table 5: Logon Details

Field	Value
User	**STUDENT##**
Password	**Training1**

   a) At the logon screen, enter the details shown in the table, Logon Details , and choose Log On .
   b) In the Google Chrome dialog box that prompts you to save your password, choose Save .
   c) In the Tips and Tricks dialog box, select the Don't show on startup checkbox and choose Close .

Task 2: Import a ready-made project and build the run-time database objects

Setting up a brand new Web IDE project requires a number of configuration steps. To keep things simple in this class, and to get started quickly, we will import a ready-made project that already has many of the settings in place. It also includes all the exercise solutions for this class.

1. Import an existing XSA project into your workspace.

The project archive, HA100.zip , is located in your HA100 folder, and must be imported to your workspace with the project name **HA100_##** (where ## is your group number).

a) Right click the Workspace folder, which is found under the Development view, and choose Import → From File System .

b) Choose Browse and navigate to the Windows folder Favorites → HA100 .

c) Choose the file HA100.zip and choose Open .

d) In the Import to field, enter **/HA100_##** (where ## is your group number).

e) Select the Extract Archive checkbox to ensure that the zip file will be unzipped during import.

f) Choose OK to begin the import.

2. Assign your project to the DEV space.

a) In the workspace, right-click your HA100_## project and choose Project Settings .

b) Open the Space panel and choose DEV from the drop down list.

c) Choose Save .

d) Choose Close .

3. Build the HDB module of your project so that the database container can be generated.

Caution:
Make sure you execute the build at the HDB module level. If you build the project, the HDB module will not be built.

a) Right-click the HA100_## → HDB folder and choose Build.

b) Wait a few moments, then check that the log shows a successful build.

4. Expand the src folder in your imported project until you reach the solutions sub–folder, where various development artifacts already exist.

a) Expand the folders HA100_## → HDB → src → solutions .

5. Open the file, CVC_US_SALES_00.hdbcalculationview , in the graphical calculation view editor , and study the definition of the calculation view.

a) In the solutions sub-folder, double-click the file CVC_US_SALES_00.hdbcalculationview and a new tab will appear. To view the definition of the calculation view, select the new tab.

Note:
You can temporarily expand any view to full screen by simply double-clicking on the tab. You can restore the view to its original size by repeating this action. This can be very helpful as many of the views contain a lot of information that is easier to read when you fill the whole screen.

6. Close the calculation view CVC_US_SALES_00.hdbcalculationview

a) Right-click on the tab and chose Close , or simply click on the 'X' to the right of the tab.

SAP HANA Cockpit

SAP HANA cockpit provides a single point of access to a range of tools for administration and monitoring of multiple, individual, and tenant SAP HANA databases. It also integrates the SQL development capabilities required by administrators. SAP HANA cockpit is a web-based HTML5 user interface that you access through a browser. It runs on SAP HANA XSA.

You can use the cockpit to monitor and manage systems running SAP HANA 2.0 or SAP HANA 1.0 SPS 12.

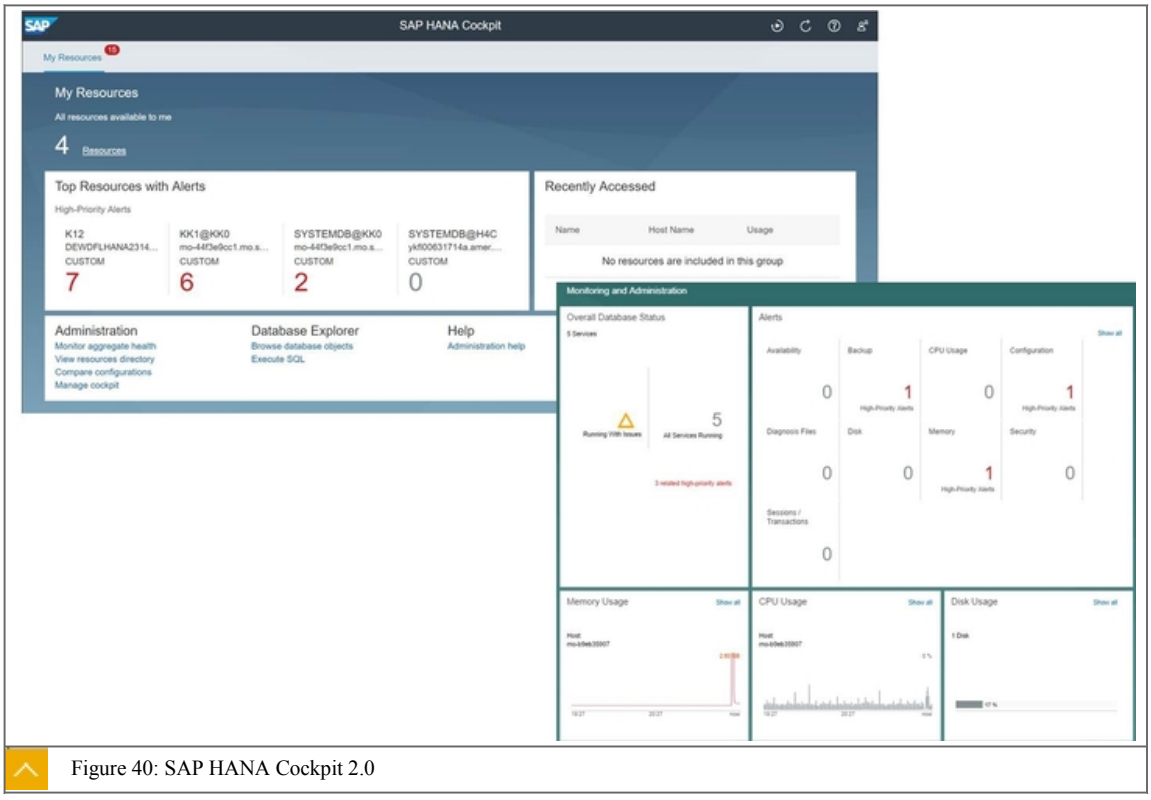


Figure 40: SAP HANA Cockpit 2.0

Advantages of SAP HANA Cockpit 2.0

The SAP HANA cockpit provides aggregate, system, and database administration features. Examples of these include database monitoring, user management, and data backup. Administrators can use the SAP HANA cockpit to start and stop services, to monitor the system, to configure system settings, and to manage users and authorizations. Cockpit pages that allow you to manage SAP HANA options and capabilities (for example, SAP HANA dynamic tiering) are only available if the option or capability has been installed.

Initially, the SAP HANA cockpit displays data at a landscape or enterprise level. You can quickly drill down to an overview of an individual resource. Links, data, tiles, and different parts of a single tile are drillable, which provides access to more detailed information and functions.

Integrated into the cockpit is the SAP HANA database explorer. The database explorer allows you to query information about the database. It does this using the in-built SQL console or MDX console, as well as the ability to view information about your database's catalog objects.

There are actually two releases of the SAP HANA Cockpit; 1.0 and 2.0. HANA 2.0 Cockpit includes all the features of HANA 1.0 Cockpit. In addition, HANA 2.0 Cockpit allows you to manage and monitor multiple HANA systems from a single location. You can manage both on-premise and cloud deployments of SAP HANA from the same cockpit.

SAP HANA Cockpit 2.0 Tasks

SAP HANA Cockpit 2.0 is a standalone component and it runs on its own HANA Express instance. It is not installed on the core SAP HANA platform. HANA Cockpit 2.0 can be used to manage multiple SAP HANA instances, both on-premise and cloud. As a comparison, SAP HANA Cockpit 1.0 was an embedded XS application that was installed on the core SAP HANA instance. It could only be used with one SAP HANA instance (the one it was installed on).

Note:
Full administration of SAP HANA is currently not possible with SAP HANA cockpit. Certain tasks continue to require SAP HANA studio. It is planned that more administration features will be moved from Studio to Cockpit 2.0 in future releases, so eventually Studio will not be required at all for administration.

HANA 2.0 Cockpit is a browser-based application. To use it, you need the latest version of Google Chrome, Mozilla Firefox, Microsoft Edge, or Apple Safari. Microsoft Internet Explorer 11 is also supported.

We recommend that for production purposes, you run the SAP HANA Cockpit 2.0 on its own physical server. However, technically it can run on the same server, side-by-side with HANA 2.0 or HANA 1.0 SPS12.

LESSON SUMMARY
You should now be able to:

- Work with SAP HANA interfaces

Unit 2
Lesson 3

Understanding Key Features of SAP HANA Database

LESSON OVERVIEW
This lesson will introduce you to the various aspects of the SAP HANA database.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand key features of SAP HANA database

Row and Column Stores

In this lesson we turn our attention to the heart of SAP HANA; the in-memory database.

SAP HANA includes a fully-relational, ACID-compliant database.

Note:
ACID is an acronym that means the database can support Atomicity, Consistency, Isolation, and Durability (ACID). This is a requirement of a database that must prove that it is 100% reliable for mission-critical applications. The database must guarantee data accuracy and integrity even when there are lots of simultaneous updates across multiple tables.

Most traditional enterprise relational database tables are row-based, as this is regarded as the optimal design for a transactional system. However, SAP HANA not only handles transactions but also analytics. For analytics, column tables are optimal. SAP HANA's database supports both row tables and column tables. Both table storage types are needed in a platform that handles transactional and analytical applications in the same database. Developers can decide which type of table they need to support their applications; that is, row or column.

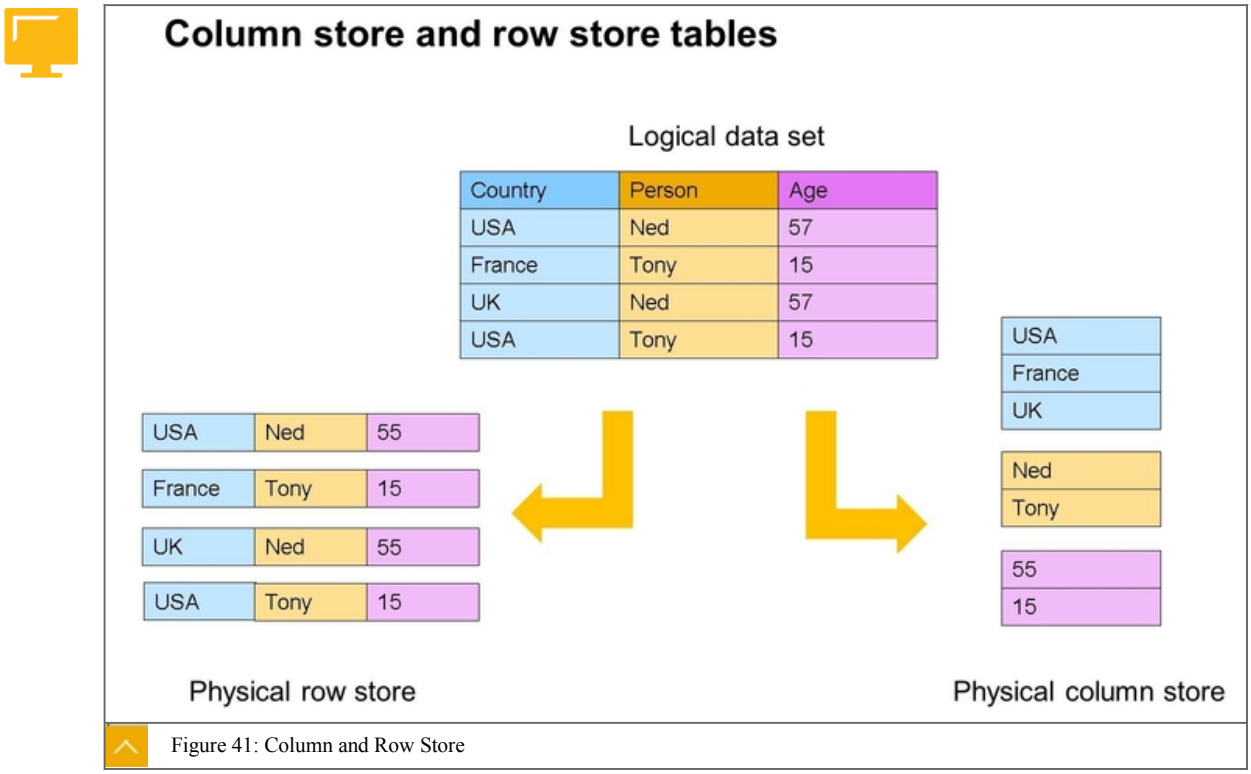


Figure 41: Column and Row Store

The figure, Column and Row Store , shows that the key difference between row and column stores is the way that the data is organized.

Column store tables are efficient for analytical applications where requests for sets of data are not predictable. Queries from analytical applications that are sent to the database often require only a subset of the overall data in the table. Usually only limited columns are required. With column store, only the required columns are loaded to memory, so you avoid using up memory with columns that will never be used. Also, the data is arranged efficiently with all values of a column appearing one after another. This continuous sequencing of the column values is preferred by the CPU, which is able to scan the values efficiently without having to skip over values.

With column store, SAP HANA scans columns of data so quickly that additional indexes are usually not required. This helps to reduce the complexity by avoiding the need to constantly create drop and rebuilding indexes.

It is easy to alter column store tables without dropping and reloading data.

Column store tables are optimal for parallel processing, as each core is able to work on a different column.

The downside to column store is the cost of reconstructing complete records from the individual column store, if all columns are required by the application. This is the case when the application is transaction-based and so, all fields are usually needed for a record update and must all be retrieved. This would be possible with column store. However, it would be slower than if the storage was row-based, where all the columns are always held together and can be read quickly.

Row storage is still needed to support transaction processing where all columns need to be retrieved. Often an application is both transactional and analytical. In this case, you must decide which is the best storage method to use. You cannot have a table that is both row and column storage. It is easy to convert a table from row to column and the other way around, and you do not lose the data when doing this.

SAP S/4HANA combines transactional and analytical applications and so, utilizes both column and row store tables.

Data Footprint Reduction

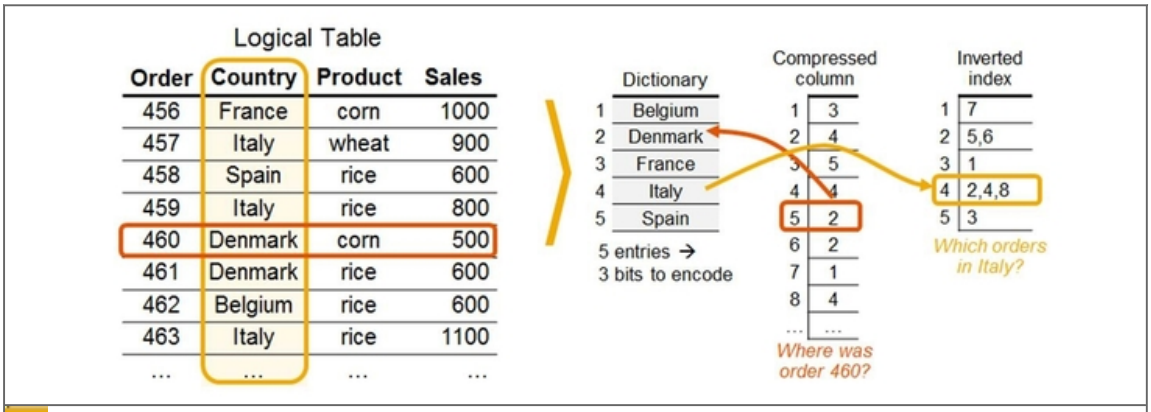


Figure 42: Compression

The data in the SAP HANA column store tables is automatically compressed. This is done to reduce the data footprint.

The following are some of the benefits of a reduced data footprint:

- You can get more data into CPU cache and therefore reduce main memory access, to maintain high performance.
- You can fit entire enterprise databases into memory and avoid disk access.
- Operations, such as backup and restore, are sped up as you deal with a smaller data size.

Mechanism of Compression

The amount by which data reduction can take place is driven by the shape of the business data. Compression is most impressive when there is a lot of repetition in the data values.

Take for example, a huge sales order table where the customer type A, B, or C is stored on each customer order. In this case, the customer type would appear a huge number of times in the column.

Compression strips out the repetition and stores each unique value once in a dictionary store. SAP HANA then uses integers to represent the business values in the original store, as this takes up far less space and is very efficient for scanning. SAP HANA links the dictionary entries to the actual table using special reference stores. These reference stores identify the position where the original value was and its corresponding business value from the dictionary store. This mechanism is embedded deep in the SAP HANA database. The processing happens invisibly.

Parallel Processing of Data

One of the key enablers of SAP HANA's incredible performance is parallel processing. With recent hardware developments, especially new multi-core processors, we can build instant-response applications by spreading the processing tasks across all the cores.

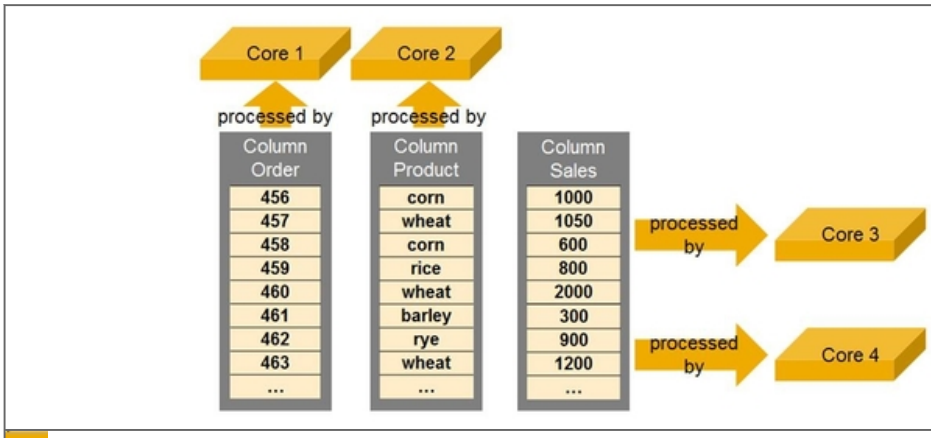


Figure 43: Parallel Processing

SAP HANA automatically spreads the workload across all cores and ensures that all parts of the hardware are contributing to the throughput.

SAP HANA is scalable. This means that you can easily add more processors as required, to increase the parallelization and therefore the speed of processing.

To take advantage of the built-in parallel processing capabilities of SAP HANA, you can use column store tables. Column store tables are automatically processed in parallel. Each column can be processed by one core. The more cores you add to the SAP HANA landscape, the more parallelization occurs.

For column store tables, you can also define partitions on each column. This means that only the required partitions are read to memory. For example, if a query requested only current-year data, then all other years in the column would be ignored. Partitions can be created based on known popular business values or by simply allowing SAP HANA to split large columns in an arbitrary way.

Note:
Up to and including SAP HANA 1.0 SPS09, the maximum number of partitions was 1,000 per column table. Since SAP HANA 1.0 SPS10 this has increased dramatically to 16,000 partitions per column table.

Parallel processing is a key enabler for real-time processing for any application powered by SAP HANA. By using column store tables with well-designed partitioning and data models, you can expect excellent performance from the database.

Data Temperatures

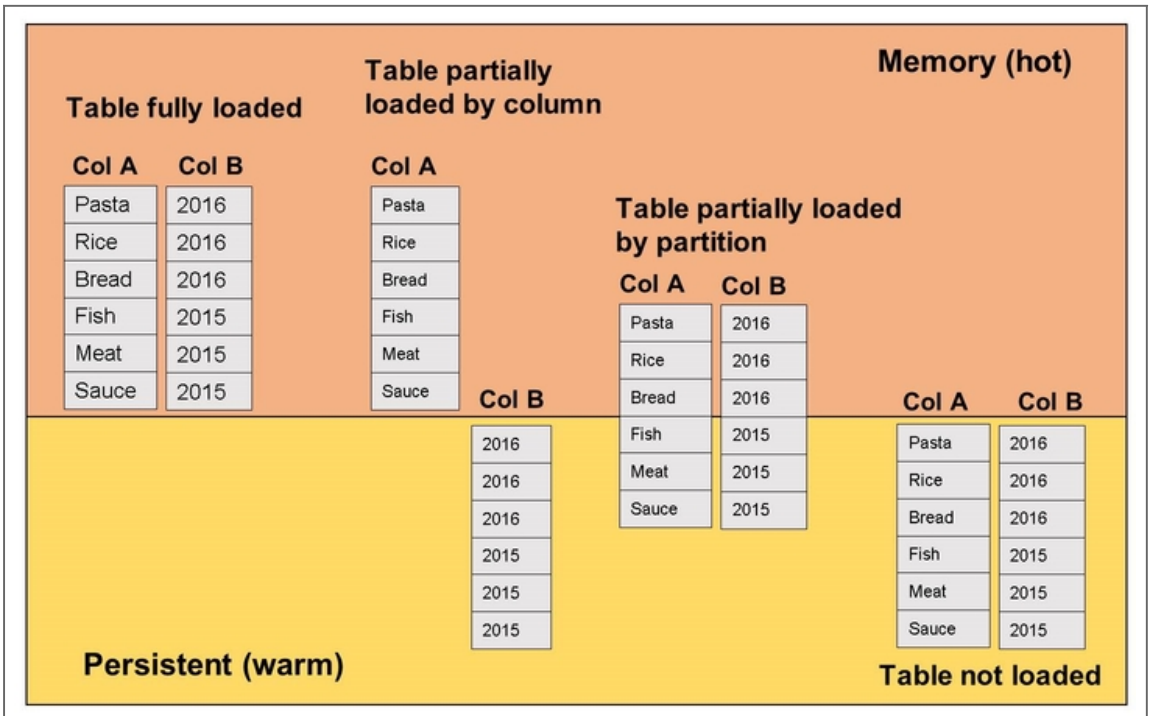


Figure 44: Data Distribution

SAP HANA is often described as an “in-memory database”. Although this is true, you must also remember that SAP HANA has a persistent storage layer too. The persistent storage layer is disk-based and is a mandatory part of SAP HANA. It is not a separate component.

There are two reasons why you need the disk layer. They are as follows:

- To provide an area to unload less important data when memory is full. We call the unloaded data “inactive data”.
- To enable data recovery if the power fails.

Note:
For now we will be focusing on the first reason. We will cover the second reason later, when we discuss high availability.

Data Distribution

It is possible to fit an entire organization’s data completely in memory if the SAP HANA memory is sized large enough. However, most organizations will size their SAP HANA system with only enough memory to hold the most useful data, and they will utilize disk to store the remaining data. This means that there will not be enough memory for the entire organization’s data and there will come a time when there may be competition for the memory. When memory is full, the data that is used less often is automatically moved from memory to disk to make way for new data. The larger the memory, the less displacement is needed. Remember also that some space is needed in memory as a working space for calculations.

Even though the cost of memory is falling and organizations can implement huge memory, it is still too expensive to be used to store all data of all ages. Data usually has a lifecycle. An organization usually values their most recent data much more highly than their older data. Generally, organizations also access their recent data much more frequently than older data.

Conceptually, data can be classified into temperatures. Data that is accessed frequently is known as hot data. Data that is accessed less frequently is known as warm data. Data that is rarely accessed (often retained only for legal purposes) is known as cold data.

Data Temperature Storage

SAP HANA handles data temperatures. It stores the hot data in memory (most expensive). Warm data is stored in the persistent disk layer (less expensive than memory). Cold data can be stored outside of SAP HANA in data archives or near-line solutions (lowest cost). For now, we will focus on hot and warm data.

It can be difficult to decide what is hot and what is warm data, and how the data moves between these layers. SAP HANA can now handle these dilemmas.

Any data that is accessed by any application always comes from memory. This means that if the table is sitting in the persistent layer, the moment it is needed, the table is automatically loaded to memory. It becomes hot data. Column tables can also be partitioned and SAP HANA can intuitively load only the required columns and partitions to memory. It leaves the unwanted columns and partitions in the persistent layer.

You can force data to be loaded from persistence to memory (and the other way around), but it is usually better to allow the applications to handle that based on real usage. Monitoring memory utilization is a key task of an administrator and there are many supplied tools that can support them. What you do not want is continual swapping between memory (hot) and disk (warm), but this will happen if memory is not sized large enough to cope.

Dynamic Tiering

We already know that the primary image of a table in SAP HANA is memory-based. This means that applications read tables from memory. If the table is not already loaded to memory (because it was displaced or never loaded), then the table is automatically loaded to memory. This is how active and inactive data is handled. however, there is another option to consider; dynamic tiering.

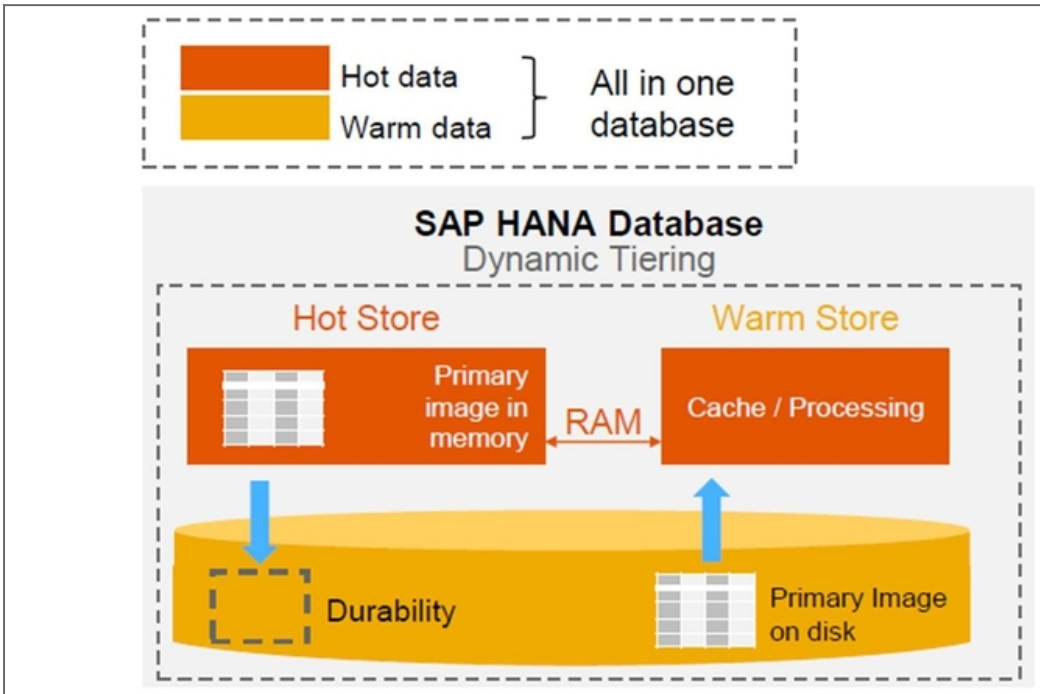


Figure 45: Dynamic Tiering

Since SAP HANA SPS09, you can implement an optional, additional database component known as extended storage. Extended storage is a disk-based storage layer that sits side by side with the existing HANA database. It is used to store tables that do not require the hyper-performance of memory-based tables. Extended storage tables work seamlessly with memory-based tables. A developer or consumer does not need to be concerned (apart from the performance aspect) with where the tables reside.

When an SAP HANA table is first created, you can decide whether it should be memory-based or it should be part of extended storage. This decision can be changed anytime, either manually or controlled by the application. When data becomes more critical, tables can be moved from extended storage to memory (and the other way around).

All tables, whether in memory or extended storage, are listed in the same SAP HANA database catalog. SAP supplies tools to allow administrators to easily monitor both storage areas.

From an architectural perspective, extended storage provides an alternative (or compliment) to Hadoop for Big Data requirements. It can be used to store data that is not subject to strict Service Level Agreements (SLA) and does not need the hyper-performance of memory-based tables.

Technically, extended storage is a column-based database, which is optimized for high-read performance. It borrows much of its technology from SAP IQ database. Although SAP HANA's memory is needed to process the extended storage data, the primary image (the complete table) is not stored in memory, and only data that is needed for each query request passes through memory.

Note:
Extended Storage is the technical name for Dynamic Tiering and is not part of core SAP HANA. It is an SAP HANA option and currently requires a separate license.

Multistore Tables

With SAP HANA 2.0, yet another option exists to support performance of the database — multistore tables.

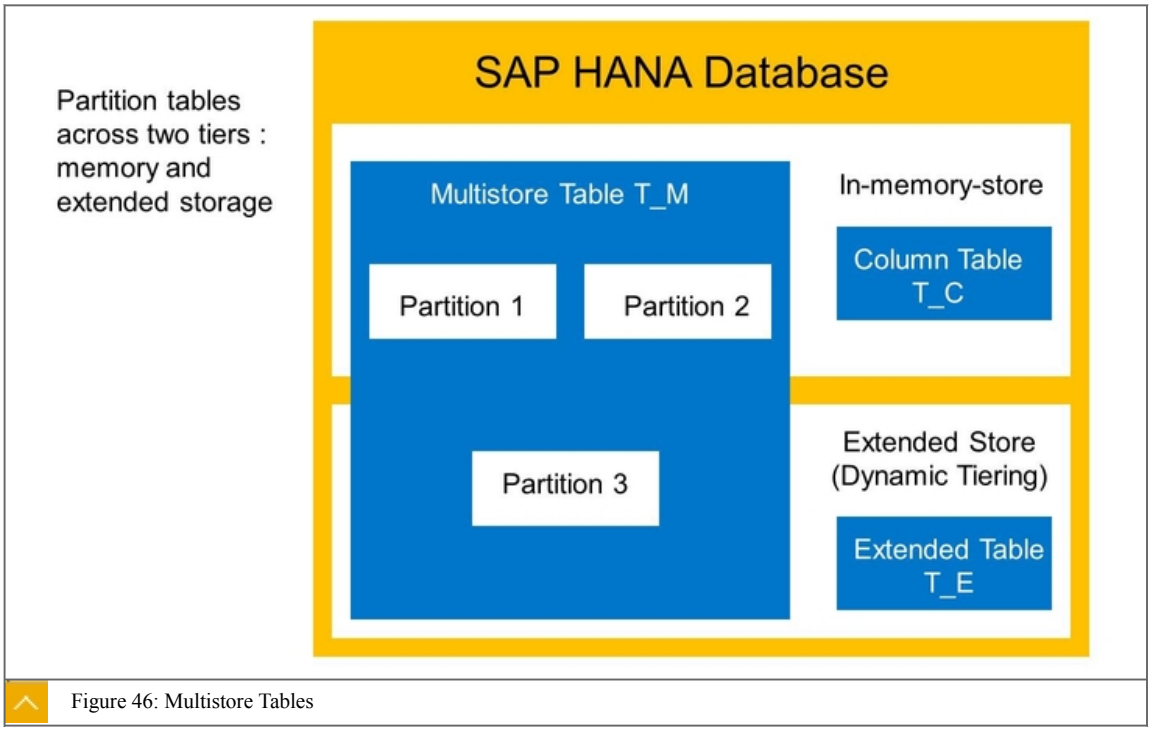


Figure 46: Multistore Tables

Multistore tables are tables that have been partitioned across two tiers — memory and extended storage (you must install extended storage to support multistore tables).

Take for example, a huge table where only the recent data in the table needs to support a strict SLA. The older data does not have to meet the SLA. Since SAP HANA 2.0, you can now have recent partitions in memory, and the remaining partitions on disk in extended storage. Before SAP HANA 2.0 you could not split partitions across the two tiers. They would either all be in memory or all be on disk in extended storage. Now you have the best of both worlds.

Active/Active-Read Enabled Mode

Since SAP HANA 1.0 SPS11, it has been possible to replicate the database log from HANA to a secondary instance of HANA and then to asynchronously replay the database log in the secondary system. This keeps the two HANAs in sync. The reason for doing this is to provide a hot, continuous backup to use in case of primary system failure. You can easily switch to the secondary system to continue with almost no disruption. Many customers have implemented this hot-standby solution.

Since SAP HANA 2.0, we have introduced active/active-read enabled mode, which builds on the solution described above.

Active/Active-Read Enabled Mode

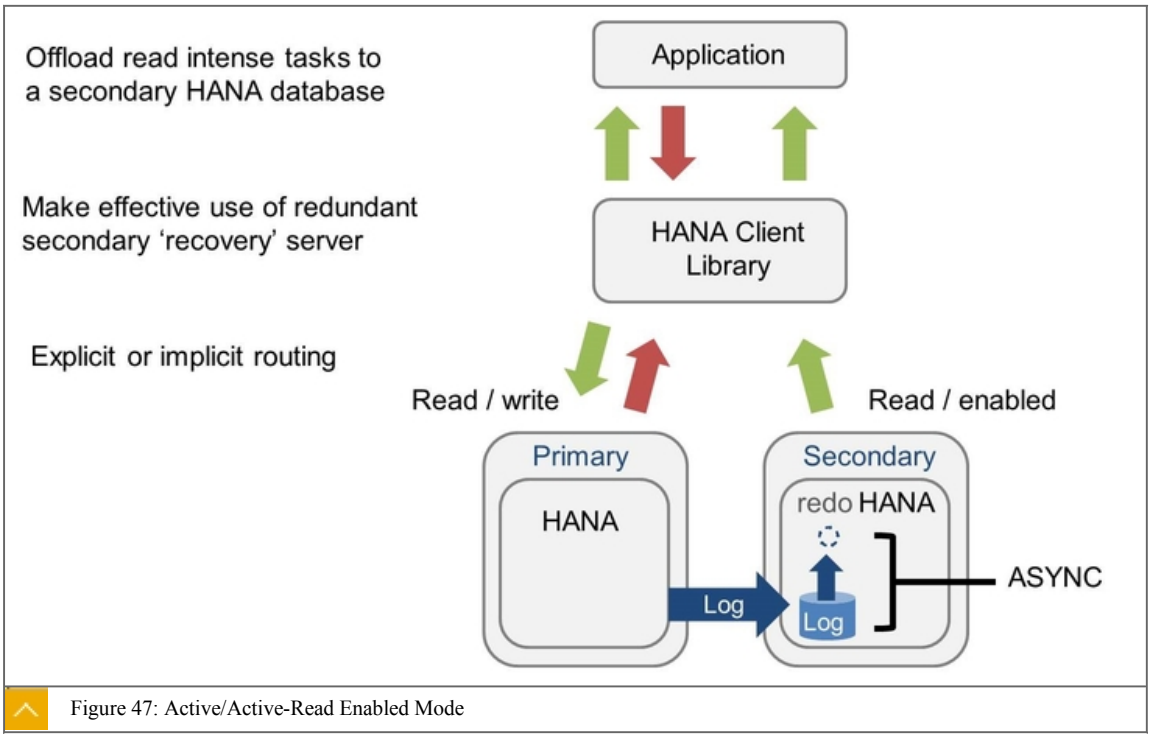


Figure 47: Active/Active-Read Enabled Mode

With active/active-read enabled mode, you can actually use the fully-synced, secondary HANA system for read-intensive tasks. This improves the overall performance of the SAP HANA database. It does this by providing a better balance of workloads, as read-intense workloads can now instead be read from the secondary system.

Applications can now be "hard-coded" to explicitly redirect read-intense activities to the secondary system, and write activities to the primary system (the two systems are automatically kept in sync). Applications can also use implicit hint-based routing to determine where the read should take place. This means that you do not hard code the routing, but allow the application to determine this dynamically.

SAP HANA Cockpit provides a side-by-side view of the performance of both systems for monitoring purposes.

Maintaining Good Read Performance with Frequent Database Updates

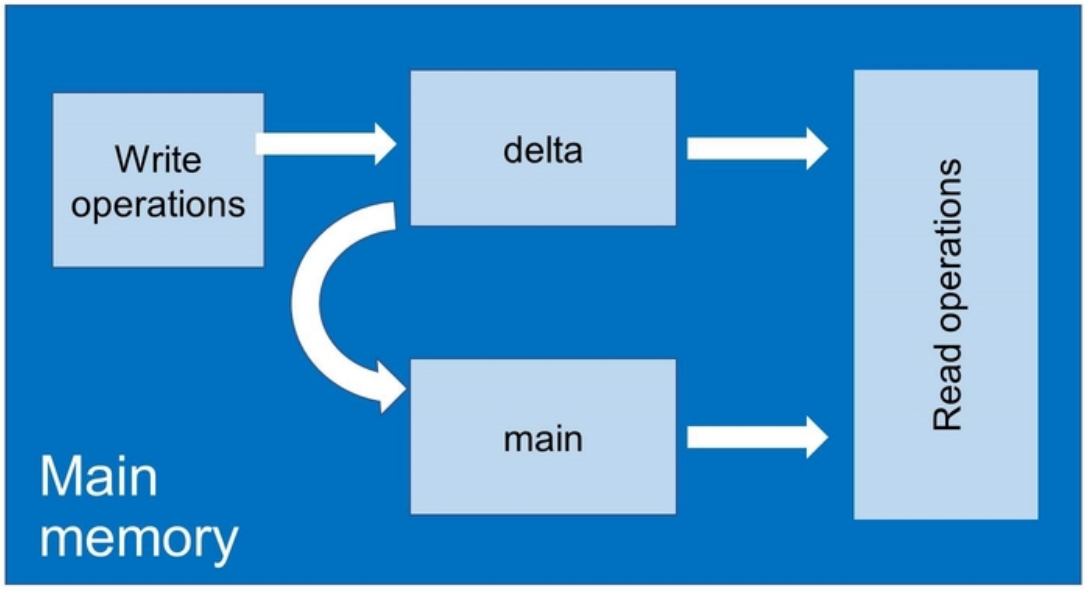


Figure 48: Delta Merge

Updating and inserting data into a compressed, sorted column store table is a costly activity. This is because each column has to be uncompressed, the new records are inserted and then recompressed again. Thus, the whole table is reorganized each time.

For this reason, SAP has separated these tables into a Main Store (read-optimized, sorted columns) and Delta Store (write-optimized, non-sorted columns or rows). There is a regular automated database activity that merges the delta stores into the main store. This activity is called Delta Merge.

Queries always run against both main and delta storage simultaneously. The main storage is the largest one, but because its data is compressed and sorted, it is also the fastest one. Delta storage is very fast for insert, but much slower for read queries, and therefore kept relatively small by running the delta merge frequently.

The delta merge can be triggered based on conditions that can be set. For example, you could define a condition that checks if the delta size is greater than 5% of the main size. If so, the delta merge is triggered.

Delta merge can also be triggered by an application. For example, SAP BW is able to suggest to SAP HANA that after a large data load, a delta merge would be a good idea. Staying on top of the delta merge is critical to maintaining good performance of SAP HANA and the administrator is responsible for this.

Note:
Refer to training course “HA200” to learn more about delta merge.

Multitenancy

SAP HANA can run multiple, isolated applications within the same system. This is called multitenancy.

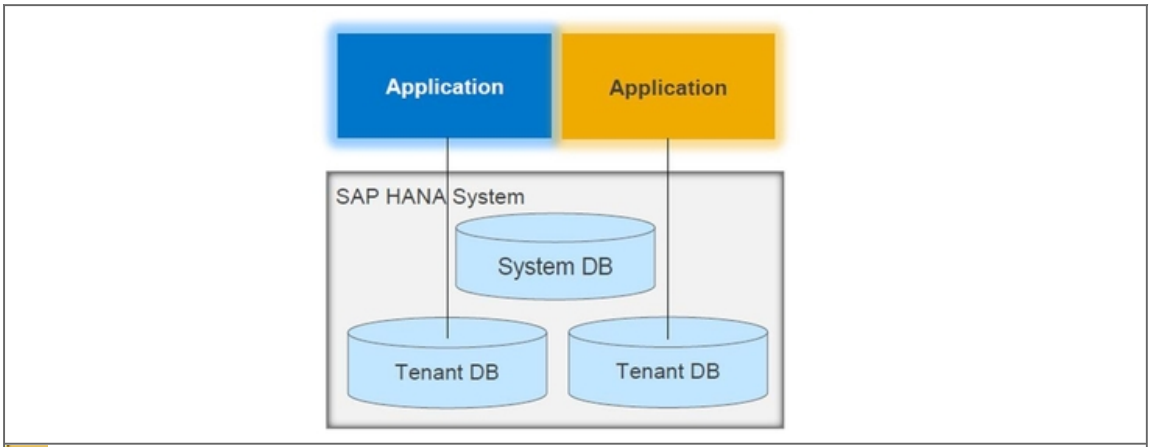


Figure 49: Multitenancy

With multitenancy, there is a strong separation of business data and users who must be kept apart. Each tenant has its own isolated database. Business users would have no idea that they are sharing a system with others running different applications.

The system layer is used to manage the system-wide settings and cross-tenant operations, such as backups.

The benefit of a multitenancy platform is that we can host multiple applications on one single SAP HANA infrastructure and share common resources to simplify and reduce costs.

Multitenancy is the basis for cost-efficient cloud computing.

Unit 2
Exercise 4

Explore the SAP HANA Database with Studio

Exercise Objectives

After completing this exercise, you will be able to do the following:

- Locate a table in the database.
- Display the metadata of a table.
- Identify the number of records in a table.
- Preview the content of a table.

Business Example

You are an IT system administrator. You need to explore the SAP HANA database.

During this exercise, you will use the following tables from the TRAINING schema:

- MARA

  This is a table that contains the product master data in an SAP ERP database.

- HANA_SEARCH_DATA

  This is a table of selected customers and their information.

1. Start the SAP HANA Studio and log on to the SAP HANA system. You can skip this step if you have already logged on.

   For this exercise, you can use either the SAP HANA Administration Console perspective or the SAP HANA Modeler perspective.

2. Locate the table MARA by using a filter on the Tables node.

3. Open the definition of table MARA and identify whether the table is row or column store.

4. Identify the key columns of the table MARA.

5. Identify the number of records loaded to the table and also the storage used by the main and delta areas.

6. Preview the data of table MARA.

7. Open the definition of table HANA_SEARCH_DATA and identify the tables storage type and indexes. Determine why there is no delta storage value for this table and why there are no partitions available.
   There is no delta storage value for this table because this is a row table, and delta storage is only relevant for column tables. There are no partitions available because this is a row table and only column tables have partitions.

Unit 2
Solution 4

Explore the SAP HANA Database with Studio

Exercise Objectives

After completing this exercise, you will be able to do the following:

- Locate a table in the database.
- Display the metadata of a table.
- Identify the number of records in a table.
- Preview the content of a table.

Business Example

You are an IT system administrator. You need to explore the SAP HANA database.

During this exercise, you will use the following tables from the TRAINING schema:

- MARA

  This is a table that contains the product master data in an SAP ERP database.

- HANA_SEARCH_DATA

  This is a table of selected customers and their information.

1. Start the SAP HANA Studio and log on to the SAP HANA system. You can skip this step if you have already logged on.

   For this exercise, you can use either the SAP HANA Administration Console perspective or the SAP HANA Modeler perspective.

2. Locate the table MARA by using a filter on the Tables node.

   a) In the Systems view, expand the Catalog folder. To view the Tables node, locate the TRAINING schema.

   b) Right-click the Tables node and choose Filters .

   c) Enter the filter pattern **MA** and choose OK.

   d) Expand the Tables node.

      The table list is now filtered and displays the table MARA in the filtered list. Note that the icon for the table represents column store, as this is a column store table.

3. Open the definition of table MARA and identify whether the table is row or column store.

   a) Right-click the table MARA and choose Open Definition .

      This screen shows the table structure with all columns, their data types, and length.

      The type of table is column store. This information is provided in the upper-right corner of the screen.

4. Identify the key columns of the table MARA.
   a) The key fields of the table are marked in the Key column.
      For table MARA, the key columns are MANDT (client) and MATNR (material number).
5. Identify the number of records loaded to the table and also the storage used by the main and delta areas.
   a) To see the number of records, choose the Runtime Information tab.
   b) Locate the Memory Consumption figures on the left side of the screen.
6. Preview the data of table MARA.
   a) Use either of the following methods to do so.
      - Right click the table name and choose Open Content to display the table content.
      - Right-click the table name and choose Open Data Preview to explore the table contents as raw data (flat table view), analyze distinct values, or build a quick analysis using a chart or a table.
7. Open the definition of table HANA_SEARCH_DATA and identify the tables storage type and indexes. Determine why there is no delta storage value for this table and why there are no partitions available.
   a) Right-click the Tables node and choose Filters .
   b) Enter the filter pattern **HANA** and choose OK.
   c) To view the filtered table list, expand the Tables node.
   d) Right-click the table HANA_SEARCH_DATA and choose Open Definition .
   e) Identify the storage type of the HANA_SEARCH_DATA table.
   f) To display the indexes created on the table, choose the Indexes tab.
   g) Verify that the table is a row-store table and that no indexes are defined.

   There is no delta storage value for this table because this is a row table, and delta storage is only relevant for column tables. There are no partitions available because this is a row table and only column tables have partitions.

Unit 2
Exercise 5

Explore the SAP HANA Database with Web IDE for SAP HANA

Exercise Objectives

After completing this exercise, you will be able to perform the following tasks:

- Locate a table in a classic database view
- Locate a synonym in a database container
- Preview the content of a table

Business Example

You are an IT system administrator. You need to explore the SAP HANA database using Web IDE for SAP HANA.

During this exercise, you will locate and explore the table PRODUCT. This is the table that contains product master data.

Task 1: Add your generated database container to the Database Explorer

1. If the SAP Web IDE for SAP HANA is not already open you must launch it and log on using the details from the table, Credentials . Otherwise you can skip this step.

   Table 6: Credentials

Field	Value
User	**STUDENT##**
Password	**Training1**

2. Switch to the Database Explorer , and when prompted, confirm that you wish to add a database to the Database Explorer . (This action simply opens the pane, so you don't have to press the '+' button, and only appears when the list is empty).
3. Now add your database container to the Database Explorer. Your database container begins with STUDENT## .
4. Explore the synonyms that have already been created for you in your database container

Note:
Remember, you have not created any local tables so there is nothing to see under 'Tables'. But we already created some synonyms for you that link to tables in an external shared schema TRAINING. You will take a look at those.

5. View the data for the table PRODUCT.

Task 2: Add a classic catalog schema based entry to the Database Explorer

In this step you will add an entry that resembles the database connection in Studio so you can directly view the database schemas and the objects they contain. This isn't always needed, as with XSA projects we work only with the container (the logical layer above the database schemas) and not directly with the database objects. But it is often helpful to a developer to have a view of the actual database too so we can see what is generated from our builds. It is also helpful to be able to see what is available in other schemas that we might want to create synonyms for in our container to allow access.

1. Add an SAP HANA Database (Multi tenant) to the Database Explorer using the information shown in the table, Database Information .

   Table 7: Database Information

Field	Value
Database Type	SAP HANA Database (Multitenant)
Host	**wdflbmt7215**
Instance Number	**00**
Database name	**H00** (Tenant Database)
User	**STUDENT##**
Password	**Training1**
Name to Show in Display	**Classic schema view**

2. Expand your database Classic schema view and view the table of contents for PRODUCT. This is found in the schema TRAINING.

Task 3: Grant access to the database objects for your application user

In HANA 2.0, database objects (e.g. tables, views, functions) are not directly accessible to an application user such as a reporting user. Access to database objects is only allowed by an internal 'technical' HANA user and this technical user is generated when a container is first built. But sometimes we need to allow an external user to also have access to the database objects when the application directly consumes a database object. An example of this scenario is when an Analysis for Microsoft Excel user tries to select a calculation view from the HANA database for their analysis. By default they will not be able to see any calculation views as the database doesn't expect to be offering its objects to this external user. What we need to do is grant permissions to the external application user also (similar to the ones the internal user has). That is what we will now do. (We need this so the exercise later will work.)

1. In the Database Explorer, open an SQL console connected to the Classic schema view database (not your own container).

   A role for enabling data access to your container's content was automatically created when you built your HDB module for the first time. This generated role (that ends in ::access role) must be granted to your predefined TRAINING_ROLE_## , which is already assigned to your application user STUDENT## . Doing this means your application user then has access to all database objects in the container's schema.

2. Grant your course participant role the authorization on your container in order to consume the container's data from external tools.

Execute the following SQL statements, which can be found in the file HA100 → SQL for External Container Access.txt . You must first replace ## with your group number.

```
GRANT "HA100_##_HDI_CONTAINER_1::access_role" to TRAINING_ROLE_##;
```

Unit 2
Solution 5

Explore the SAP HANA Database with Web IDE for SAP HANA

Exercise Objectives

After completing this exercise, you will be able to perform the following tasks:

- Locate a table in a classic database view
- Locate a synonym in a database container
- Preview the content of a table

Business Example

You are an IT system administrator. You need to explore the SAP HANA database using Web IDE for SAP HANA.

During this exercise, you will locate and explore the table PRODUCT. This is the table that contains product master data.

Task 1: Add your generated database container to the Database Explorer

1. If the SAP Web IDE for SAP HANA is not already open you must launch it and log on using the details from the table, Credentials . Otherwise you can skip this step.

   Table 6: Credentials

Field	Value
User	**STUDENT##**
Password	**Training1**

   a) Double-click the file Web IDE for SAP HANA in the folder Favorites → HA100 → URLs.

   b) At the logon screen, enter the credentials from the table, Credentials , and choose Log On.

   c) Close the Tips and Tricks dialog box that appears.

2. Switch to the Database Explorer , and when prompted, confirm that you wish to add a database to the Database Explorer . (This action simply opens the pane, so you don't have to press the '+' button, and only appears when the list is empty).

   a) On the left of the screen you will notice three buttons in a vertical arrangement. Select the middle button that looks like a set of database tables.

   b) At the prompt, choose Yes .

3. Now add your database container to the Database Explorer. Your database container begins with STUDENT## .

a) Choose the + icon on the toolbar.

b) Set the Database Type to HDI Container .

c) In the list of HDI containers, highlight your container.
Your container is the one that begins with STUDENT## .

d) For the Name to Show in Display , enter **`My HA100 Container`**.

e) Choose OK.
Your container should now appear on the top-left pane.

4. Explore the synonyms that have already been created for you in your database container

Note:
Remember, you have not created any local tables so there is nothing to see under 'Tables'. But we already created some synonyms for you that link to tables in an external shared schema TRAINING. You will take a look at those.

a) Expand your container and you should see a list of database objects types.

b) Choose Synonyms .
You should see, in the lower pane, a list of all synonyms that expose all tables needed in this training.

5. View the data for the table PRODUCT.

a) Right-click the synonym HA100 :: PRODUCT and choose Open Data .
You should see the content of the table.

Task 2: Add a classic catalog schema based entry to the Database Explorer

In this step you will add an entry that resembles the database connection in Studio so you can directly view the database schemas and the objects they contain. This isn't always needed, as with XSA projects we work only with the container (the logical layer above the database schemas) and not directly with the database objects. But it is often helpful to a developer to have a view of the actual database too so we can see what is generated from our builds. It is also helpful to be able to see what is available in other schemas that we might want to create synonyms for in our container to allow access.

1. Add an SAP HANA Database (Multi tenant) to the Database Explorer using the information shown in the table, Database Information .

Table 7: Database Information

Field	Value
Database Type	SAP HANA Database (Multitenant)
Host	**`wdflbmt7215`**
Instance Number	**`00`**
Database name	**`H00`** (Tenant Database)
User	**`STUDENT##`**
Password	**`Training1`**

Field	Value
Name to Show in Display	**Classic schema view**

a) In the Database type dropdown list, choose SAP HANA Database (Multitenant) .

b) Enter the information shown in the table, Database Information .

c) Choose OK and the database now appears in the top pane.
If needed, choose the Refresh button.

2. Expand your database Classic schema view and view the table of contents for PRODUCT. This is found in the schema TRAINING.

a) Expand the database Classic schema view and then expand the Catalog node.
You should see a list of schemas.

b) Expand the schema TRAININGand choose Tables .
You should see a list of tables on the pane below.

c) Right-click the table PRODUCT. To view the table of contents, choose Open Data .

Task 3: Grant access to the database objects for your application user

In HANA 2.0, database objects (e.g. tables, views, functions) are not directly accessible to an application user such as a reporting user. Access to database objects is only allowed by an internal 'technical' HANA user and this technical user is generated when a container is first built. But sometimes we need to allow an external user to also have access to the database objects when the application directly consumes a database object. An example of this scenario is when an Analysis for Microsoft Excel user tries to select a calculation view from the HANA database for their analysis. By default they will not be able to see any calculation views as the database doesn't expect to be offering its objects to this external user. What we need to do is grant permissions to the external application user also (similar to the ones the internal user has). That is what we will now do. (We need this so the exercise later will work.)

1. In the Database Explorer, open an SQL console connected to the Classic schema view database (not your own container).

A role for enabling data access to your container's content was automatically created when you built your HDB module for the first time. This generated role (that ends in ::access role) must be granted to your predefined TRAINING_ROLE_## , which is already assigned to your application user STUDENT## . Doing this means your application user then has access to all database objects in the container's schema.

a) Choose Tools → Database Explorer .

b) In the Database/Containers list on the left, choose the Classic schema view entry.

c) Choose the Open SQL Console icon.

2. Grant your course participant role the authorization on your container in order to consume the container's data from external tools.

Execute the following SQL statements, which can be found in the file HA100 → SQL for External Container Access.txt . You must first replace ## with your group number.

```
GRANT "HA100_##_HDI_CONTAINER_1::access_role" to TRAINING_ROLE_##;
```

a) In Windows Explorer, open the file HA100 → SQL for External Container Access.txt .

b) Copy and paste the content of this file to the SQL Console in the Web IDE.

c) To replace ## with your group number everywhere in the SQL statements, press **Ctrl** +**H**.

d) Execute the SQL statement by pressing **F8**.

You can test if this worked because under the catalog node of your Classic schema view you should now see your container HA100_##_HDI_CONTAINER_1 , and the database objects it contains appear.

LESSON SUMMARY
You should now be able to:

- Understand key features of SAP HANA database

Unit 2
Lesson 4

Understanding Run-Time and Design-Time Architectures

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand run-time and design-time architectures

Run-Time and Design-Time Architectures

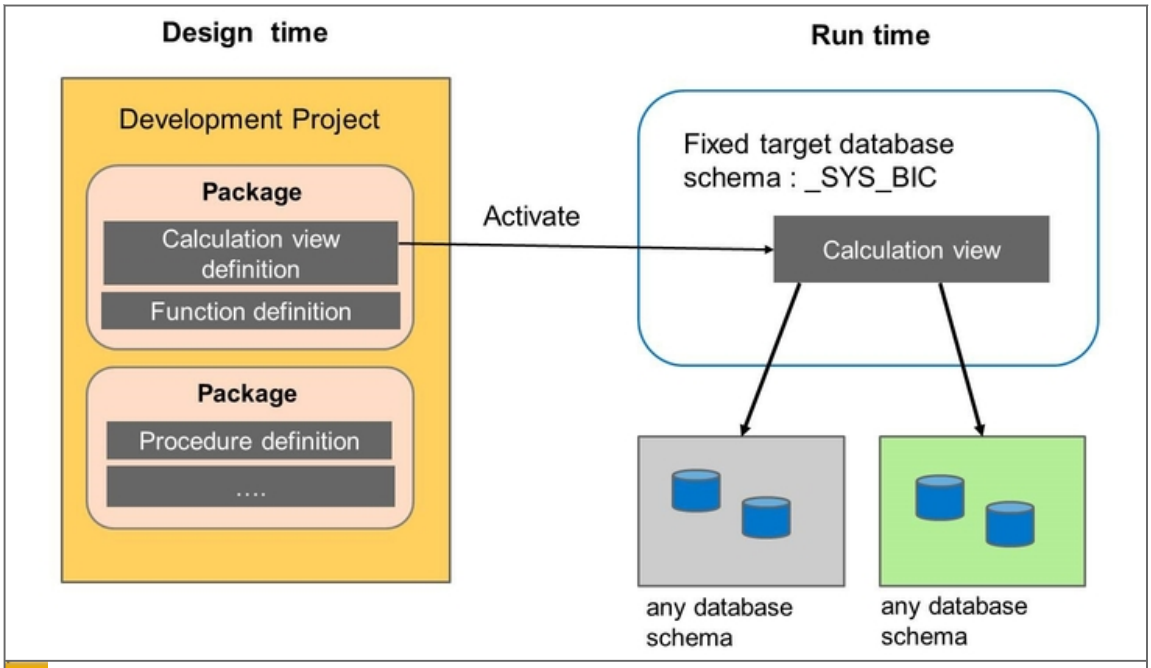


Figure 50: Simple Architecture of XS / Classic Repository

Since SAP HANA 1.0 was released in early 2011, the design and run-time architecture has remained stable.

Database artifacts and source code would sit in a giant single system-wide repository that was organized by packages. From there, development artifacts would be activated, and the generated run-time database objects and applications would be placed in database schemas and in the XS engine run-time. Only one version of the activated objects could exist, which meant you could not deploy an updated version of an application whilst still retaining the original version. XS was tightly integrated with the HANA database, which meant you could not scale either separately.

Each application that was developed could only have one deployment target. For example, you could not develop a single application and deploy it as both an on-premise and a cloud

application. You would have to create two separate applications. This original architecture is known as “XS classic/repository based”.

Since SAP HANA 1.0 SPS11, a new, flexible approach that is much more powerful has been introduced for development and run-time. Customers are encouraged to move over to the new architecture to take advantage of many new features. Eventually, XS classic will no longer be relevant and will not be the focus of any more development by SAP.

Flexible Architecture of XSA/HDI

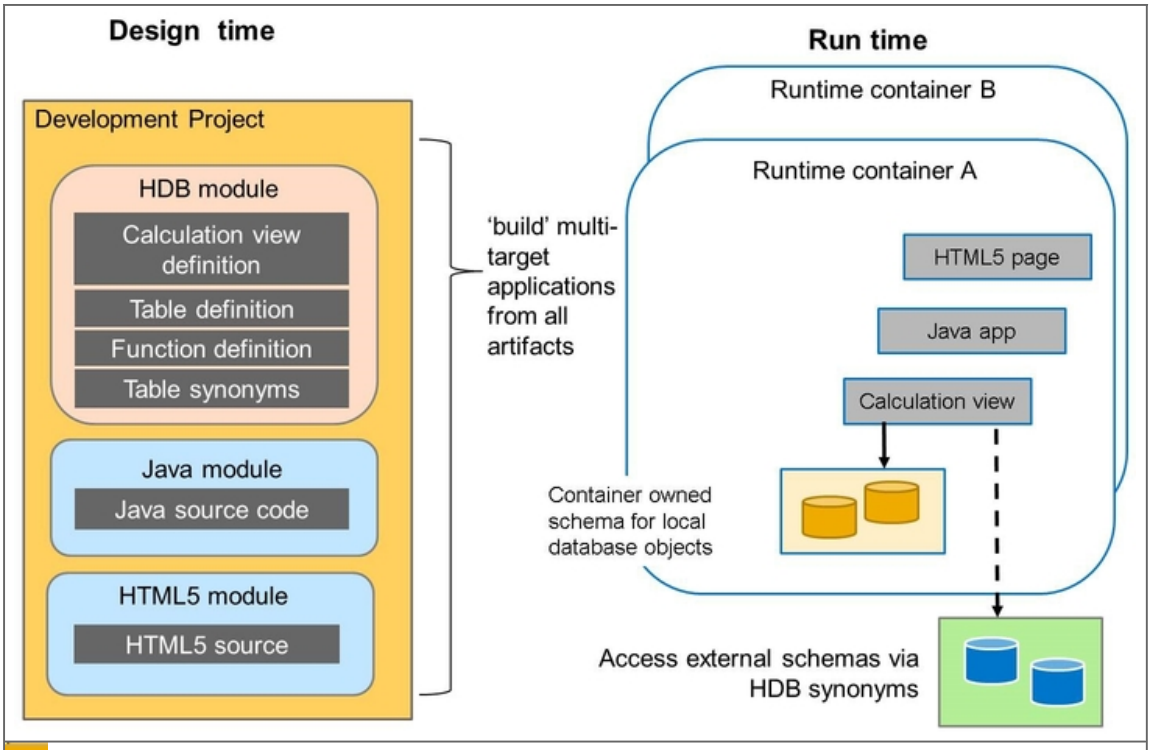


Figure 51: Flexible Architecture of XSA/HDI

The new architecture includes a more advanced version of Extended Services (XS), called Extended Services Advanced (XSA). XSA is decoupled from the database to provide more flexibility and ease of scalability. XSA can now be sized independently from the HANA database.

Application source code is no longer stored in the internal repository of HANA, but in an industry standard, external repository called Git. This allows much improved version control and collaboration. Git manages source objects across multiple instances of HANA.

For managing database objects, the new architecture includes a new deployment approach called HANA Deployment Infrastructure (HDI). This includes the generation of deployment containers, which are used to store all run-time objects for an entire application.

SAP Web IDE for SAP HANA

A new web-based development client has been introduced to support XSA and HDI. This is called SAP Web IDE for SAP HANA.

With this new architecture you can build multitarget applications (MTAs). This means that one development project can be deployed multiple times to on-premise or cloud applications. It also means that you can run different versions of the same applications simultaneously without having to have separate copies of the development projects.

Applications that run on the classic infrastructure do not run on the new architecture. SAP HANA 2.0 can support classic and the new advanced architectures side by side, so customers can run legacy applications on the classic architecture and also develop new ones in parallel on the new architecture.

Note:
Additional information relating to the new XSA/HDI architecture is covered in the HA100 Unit 5 “Powering Applications with SAP HANA”.

LESSON SUMMARY
You should now be able to:

- Understand run-time and design-time architectures

Unit 2
Lesson 5

Describing High Availability

LESSON OVERVIEW
In this lesson we will explain how SAP HANA keeps running during power outages.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe high availability

How SAP HANA Keeps Going when Servers Fail

SAP HANA utilizes memory for storage, but once the power is gone, all of the data in memory is lost.

SAP HANA must ensure that you do not lose data when the power goes.

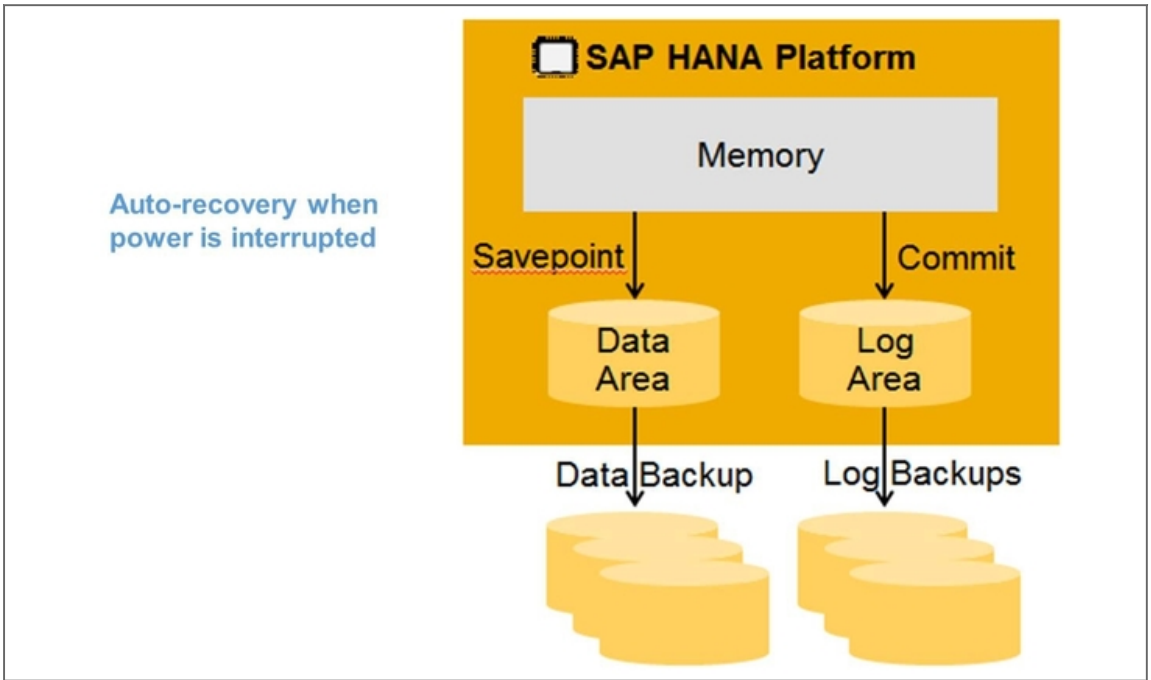


Figure 52: High Availability: Power Interruption

Every few minutes, SAP HANA automatically takes a snapshot of the entire memory. It stores this snapshot on the disk layer in an area called the data volume. This is called a savepoint. The frequency of savepoints is configurable and it depends on how frequently the database changes due to updates, inserts, and deletes. When power is restored, SAP HANA automatically reloads memory from the last savepoint. It is possible to collect many savepoints over time so that a restore can take place from any point in time.

However, it is important to develop a mechanism to ensure that no data is lost, even between savepoints. To do this, every committed transaction is recorded and saved to a log area. This

log area is often based on flash memory (SSD) to ensure ultrafast access. Thus, every update to the database since the last savepoint is captured.

High Availability: Power Interruption

When power is restored, SAP HANA automatically reads the last savepoint and then reapplies the committed transactions from the log since the savepoint. This ensures that the system is exactly where it was at the precise moment you lost the power.

This all happens automatically in the background.

Even though a complete reload of memory from the persistent layer is fast, it is also possible to identify the most important tables so that they are loaded to memory ahead of the tables that are less important. This allows you to restore great performance as quickly as possible.

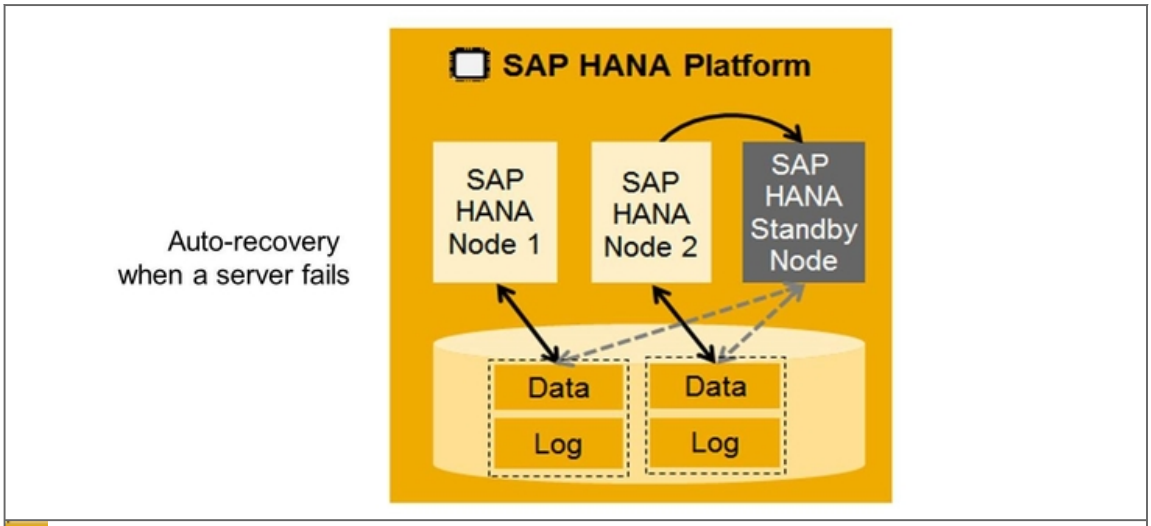


Figure 53: High Availability: Server Down

SAP HANA can be installed across multiple nodes. This is called scale-out. Scale-out is often used to spread the processing load across multiple servers, which improves performance. Scale-out is also used to provide redundant servers that are on standby in case active servers fail.

If a server fails, SAP HANA can automatically swap out to a standby server to ensure that downtime is minimized, or even eliminated.

A standby server can be on warm standby, which means that it is in a near-ready state and does not need to be started from cold.

Standby Servers

Standby servers can also be on hot standby. In this case, the primary server replicates the database log in real-time to a secondary server. This secondary server continuously replays the database log so that the databases are always in sync. This means that there is almost no downtime when switching to the standby server, as it is identical to the primary server at all times. This approach would be necessary for a mission-critical operation where down-time would be harmful to the business.

Note:
The secondary server can also be used in an active/active-read enabled mode. This means that not only is the secondary server used in case the primary server fails, but the secondary server is also used to offload all read-intense work away from the primary server to balance the workloads.

For standby servers that are not running in hot standby mode, SAP HANA uses the savepoints and logs, described earlier, to load the standby server with the latest data. This means that more time is taken to bring the servers up than hot standby.

So, for mission critical applications, and where SLAs are implemented, you can ensure customers' systems are always running by implementing these "fail-over" solutions.

Note:
For more information on all high-availability, refer to course "HA200".

LESSON SUMMARY
You should now be able to:

- Describe high availability

Unit 2
Lesson 6

Describing Security Features of SAP HANA

LESSON OVERVIEW
This lesson covers the key security features of SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe security features of SAP HANA

Key SAP HANA Security Features

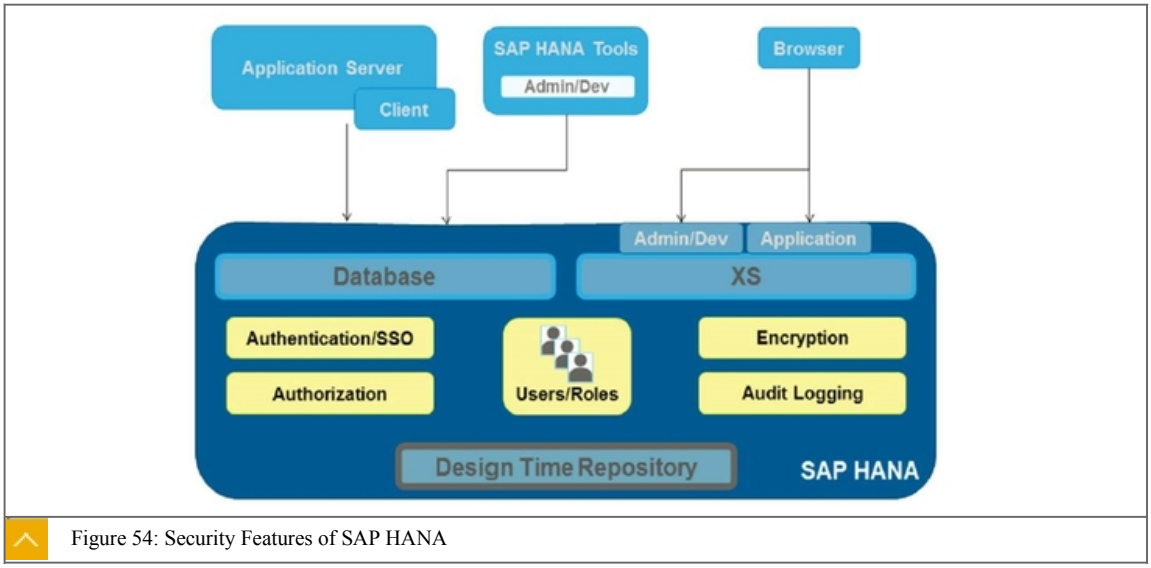


Figure 54: Security Features of SAP HANA

SAP HANA provides many crucial features to ensure that the database, XS applications (advanced and classic), and development objects in the design-time repositories are thoroughly secure. At the same time, SAP HANA offers a wide variety of access options.

Some of the key features of SAP HANA Security are as follows:

- Authentication

  No user or application is allowed to consume content from SAP HANA without first providing validated credentials. SAP HANA can handle many forms of authentication including user and password entry or various single sign-on techniques using industry standards (Kerberos, SAML). SAP HANA can also communicate with various LDAP providers, like Microsoft Active Directory, to authenticate users who are part of LDAP Groups.

- Users/Roles

SAP HANA requires you to define roles and users and then assign them to each other. This is done to create an easy to maintain authorization concept that is reusable. Roles can inherit other roles to form a hierarchy. There are many role templates provided by SAP that align to well-known developer and administrator job roles.

- Authorization

  It is possible to grant and revoke authorization for each role or user using Analytic Privileges. This determines whether users have access to business data, database objects, system actions, development objects, projects, and more.

- Encryption

  Data encryption ensures that not only your savepoints but also logs in the persistent layer are stored securely. Encryption also ensures secure communication between SAP HANA components.

- Audit Logging

  SAP HANA keeps track of all accesses to any resources, data, and functions as well as all administration of users, roles, security parameters, and rules. The logged data can be stored securely outside of SAP HANA to ensure even SAP HANA administrators cannot access this data.

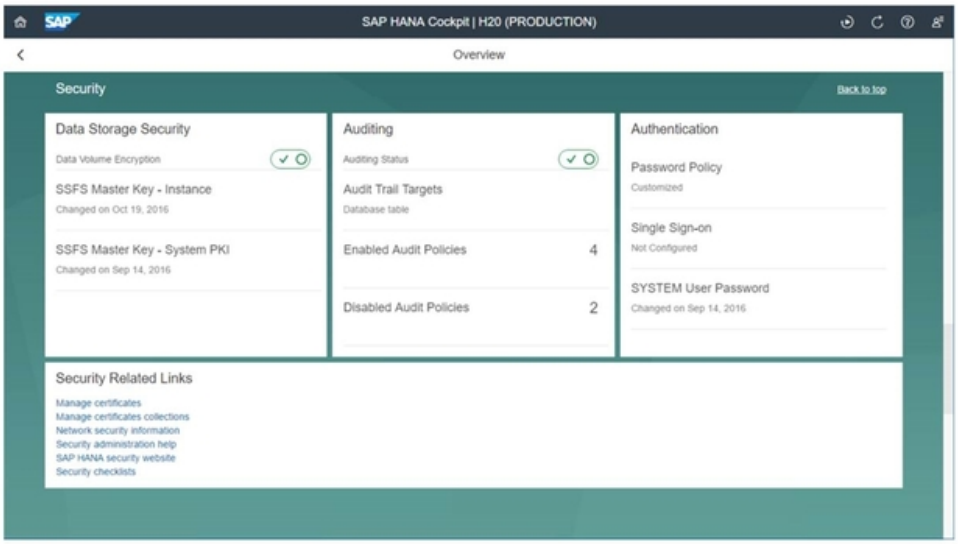


Figure 55: Managing Security with SAP HANA Cockpit

Security is managed from the SAP HANA Cockpit, where it is possible to obtain an overview of all key security parameters and drill down to details.

Before the arrival of the SAP HANA Cockpit, security was managed from Studio. It is still possible to perform many of the security-related tasks, such as creating a user or role, or setting passwords, in Studio. However, we recommend that you use the SAP HANA Cockpit as this is the focus of security administration going forward.

Using the SAP HANA Cockpit, it is possible to manage security across a number of SAP HANA instances.

Note:
For detailed information on how to set up security in SAP HANA please refer to the training course “HA240”.

LESSON SUMMARY
You should now be able to:

- Describe security features of SAP HANA

Unit 2

Learning Assessment

1. Where is SDI used?

   Choose the correct answers.

   - ☐ **A** Data cleansing
   - ☐ **B** Data transformation
   - ☐ **C** Data streaming
   - ☐ **D** For data loading

2. How many processing tiers do you usually find in a native SAP HANA application stack?

   Choose the correct answer.

   - ☐ **A** 1
   - ☐ **B** 2
   - ☐ **C** 3

3. What is XS?

   Choose the correct answer.

   - ☐ **A** A native, lightweight application server
   - ☐ **B** An optional engine to handle excessive data loads
   - ☐ **C** A data cleansing engine
   - ☐ **D** A user interface

4. Why did SAP move from XS to XSA?

   Choose the correct answers.

   - ☐ **A** To move to Cloud Foundry architecture to allow flexible application deployment options
   - ☐ **B** To include ABAP as a key development language for native SAP HANA applications
   - ☐ **C** To implement micro-services architecture
   - ☐ **D** To provide better source code management tools inside SAP HANA

5. What are the two views in the Web IDE for SAP HANA?

   Choose the correct answers.

   - ☐ **A** Database Explorer
   - ☐ **B** Development
   - ☐ **C** Catalog

6. In SAP HANA Studio, what would you find in a Package?

   Choose the correct answers.

   - ☐ **A** Calculation Views
   - ☐ **B** Analytic Privileges
   - ☐ **C** Procedures
   - ☐ **D** Tables

7. In Web IDE for SAP HANA, what can I create in a HDB module?

   Choose the correct answers.

   - ☐ **A** Calculation View
   - ☐ **B** Analytic Privilege
   - ☐ **C** HTML file
   - ☐ **D** Flowgraph

8. What is a role of the SAP HANA Cockpit?

   Choose the correct answer.

   - ☐ **A** To provide KPIs to help focus on business performance
   - ☐ **B** To provide key system performance information for one or more HANA instances
   - ☐ **C** To provide an application developer collaboration hub

9. What are advantages of column store tables?

   Choose the correct answers.

   - ☐ **A** Data footprint is automatically reduced through compression
   - ☐ **B** Only the columns required are actually loaded to memory
   - ☐ **C** Columns can be partitioned
   - ☐ **D** They are optimized for high-volume read/write transactional processing

10. Row store tables are more efficient when there is a lot of repeating data values in columns

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

11. Why do we still need a persistent layer?

    Choose the correct answers.

    - ☐ **A** To store data that has been unloaded from memory
    - ☐ **B** To hold the delta store for newly-arrived records
    - ☐ **C** To enable full database recovery if we have a power failure
    - ☐ **D** To store data that is frequently used

12. What is a multistore table?

    Choose the correct answer.

    - ☐ **A** A table that can hold data in row and column orientation
    - ☐ **B** A table that can distribute its partitions across memory and extended storage tiers
    - ☐ **C** A table that can be shared across multiple instances of HANA

13. Active/active-read enabled mode allows applications to redirect read-intense workloads to a secondary HANA?

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

14. What is a multi-target application (MTA)?

    Choose the correct answer.

    - ☐ **A** An application built with SAP HANA XS Classic (XS) that supports on-premise and cloud deployments
    - ☐ **B** An application built with SAP HANA XS Advanced (XSA) that supports on-premise and cloud deployments

15. What are the two storage components used to restore the database in case of power failure?

    Choose the correct answers.

    - ☐ **A** Log area
    - ☐ **B** Data volume
    - ☐ **C** Memory
    - ☐ **D** Delta store

16. What is scale-out?

    Choose the correct answers.

    - ☐ **A** Use of standby servers in the event of hardware failure
    - ☐ **B** Use of remote servers to store archived data that is rarely used
    - ☐ **C** Use of commodity servers that are used in high volume steaming applications
    - ☐ **D** Use of multiple servers to spread processing and improve performance

17. SAP HANA security services include which of the following?

Choose the correct answers.

- ☐ **A** Audit logging
- ☐ **B** Authentication
- ☐ **C** Encryption
- ☐ **D** Antivirus Protection
- ☐ **E** Authorization

Unit 2

Learning Assessment - Answers

1. Where is SDI used?

   Choose the correct answers.

   [ ] **A** Data cleansing

   [X] **B** Data transformation

   [ ] **C** Data streaming

   [X] **D** For data loading

   You are correct! SDI is the in-built ETL component of SAP HANA and is used for data transformation and loading — SDQ is for cleansing — SDS is for streaming. Refer to HA100 Unit 2 Architecture of SAP HANA for details.

2. How many processing tiers do you usually find in a native SAP HANA application stack?

   Choose the correct answer.

   [ ] **A** 1

   [X] **B** 2

   [ ] **C** 3

   You are correct! In a native SAP HANA application (built with XS/XSA) we need only the presentation layer and SAP HANA. SAP HANA provides both the application services and also the database services. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

3. What is XS?

   Choose the correct answer.

   [X] **A** A native, lightweight application server

   [ ] **B** An optional engine to handle excessive data loads

   [ ] **C** A data cleansing engine

   [ ] **D** A user interface

   You are correct! XS does not cleanse data, it is not an interface, and it does not handle excessive data loads — XS is an lightweight application server. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

4. Why did SAP move from XS to XSA?

   Choose the correct answers.

   [X] **A** To move to Cloud Foundry architecture to allow flexible application deployment options

   [ ] **B** To include ABAP as a key development language for native SAP HANA applications

   [X] **C** To implement micro-services architecture

   [ ] **D** To provide better source code management tools inside SAP HANA

   You are correct! XSA uses Cloud Foundry architecture. ABAP is not supported by XSA. Micros-service architecture is supported by XSA. Source code is no longer managed by SAP HANA when using XSA. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

5. What are the two views in the Web IDE for SAP HANA?

   Choose the correct answers.

   [X] **A** Database Explorer

   [X] **B** Development

   [ ] **C** Catalog

   You are correct! Database Explorer and Development are both views in Web IDE. Refer to HA100 Unit 2, Architecture of SAP HANA for more details.

6. In SAP HANA Studio, what would you find in a Package?

   Choose the correct answers.

   [X] **A** Calculation Views

   [X] **B** Analytic Privileges

   [X] **C** Procedures

   [ ] **D** Tables

   You are correct! In a HANA Studio package you will find Calculation Views, Analytic Privileges, and Procedures but not tables. Tables are found in the schemas, which are located under the catalog node. Refer to HA100 Unit 2, Architecture of SAP HANA for more details.

7. In Web IDE for SAP HANA, what can I create in a HDB module?

   Choose the correct answers.

   [X] **A** Calculation View

   [X] **B** Analytic Privilege

   [ ] **C** HTML file

   [X] **D** Flowgraph

   You are correct! HDB modules are used to build modeling objects such as Calculation Views, Analytic Privileges, and Flowgraphs.. A HTML file is not a modeling object, it is used to generate a browser page. Refer to HA100, Unit 2, Architecture of SAP HANA for more details.

8. What is a role of the SAP HANA Cockpit?

   Choose the correct answer.

   [ ] **A** To provide KPIs to help focus on business performance

   [X] **B** To provide key system performance information for one or more HANA instances

   [ ] **C** To provide an application developer collaboration hub

   You are correct! SAP HANA Cockpit provides key HANA system performance information as well as many other administration tools. Refer to HA100 Unit 2, Architecture of SAP HANA for more details.

9. What are advantages of column store tables?

   Choose the correct answers.

   [X] **A** Data footprint is automatically reduced through compression

   [X] **B** Only the columns required are actually loaded to memory

   [X] **C** Columns can be partitioned

   [ ] **D** They are optimized for high-volume read/write transactional processing

   You are correct! Column store tables are optimized for read-intense analytical processing, not transactional processing where read and write is needed. They also use automatic compression to reduce the footprint and we only need to load columns that are required into memory. Also, you can partition column tables. You can’t do any of these things with row tables. Refer to HA100 Unit 2, Architecture of SAP HANA , for details.

10. Row store tables are more efficient when there is a lot of repeating data values in columns

    Determine whether this statement is true or false.

    [ ] True

    [X] False

    You are correct! Column tables are more efficient when there is lots of repeating data values in columns. This is because we can remove all duplicates of each value and in place use binary integers for a smaller footprint and faster processing. Refer to HA100, Unit 2, Architecture of SAP HANA , for details.

11. Why do we still need a persistent layer?

    Choose the correct answers.

    [X] **A** To store data that has been unloaded from memory

    [ ] **B** To hold the delta store for newly-arrived records

    [X] **C** To enable full database recovery if we have a power failure

    [ ] **D** To store data that is frequently used

    You are correct! The persistent layer is needed to off-load low priority data from memory to disk when memory is full. It is also used to store periodic snapshots of memory in case we lose power and need to recover memory from the last saved snapshot. Refer to HA100, Unit 2, Architecture of SAP HANA , for details.

12. What is a multistore table?

    Choose the correct answer.

    [ ] **A** A table that can hold data in row and column orientation

    [X] **B** A table that can distribute its partitions across memory and extended storage tiers

    [ ] **C** A table that can be shared across multiple instances of HANA

    You are correct! A multistore table is a new feature of SAP HANA 2.0, where partitions can be spread across memory and extended storage. Refer to HA100, Unit 2, Architecture of SAP HANA, for details.

13. Active/active-read enabled mode allows applications to redirect read-intense workloads to a secondary HANA?

    Determine whether this statement is true or false.

    [X] True

    [ ] False

    You are correct! A secondary read-only HANA is used to continuously capture and replay the database log from a primary HANA. The secondary read-only HANA can then be used for any read-intense tasks, thereby off-loading this work from the primary HANA to balance the workloads and increase overall performance. Refer to HA100, Unit 2, Architecture of SAP HANA , for details.

14. What is a multi-target application (MTA)?

    Choose the correct answer.

    [ ] **A** An application built with SAP HANA XS Classic (XS) that supports on-premise and cloud deployments

    [X] **B** An application built with SAP HANA XS Advanced (XSA) that supports on-premise and cloud deployments

    You are correct! A multi-target application (MTA) is built with XS Advanced and allows single development projects to be deployed to on premise or cloud. With XS you cannot build MTAs. Refer to Unit 2, Architecture of SAP HANA for details.

15. What are the two storage components used to restore the database in case of power failure?

    Choose the correct answers.

    [X] **A** Log area

    [X] **B** Data volume

    [ ] **C** Memory

    [ ] **D** Delta store

    You are correct! The two components used are data volume, to store the savepoints that periodically capture snapshots of memory, and log area, to store the log that replays all committed transactions since the last savepoint. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

16. What is scale-out?

Choose the correct answers.

[X] **A** Use of standby servers in the event of hardware failure

[ ] **B** Use of remote servers to store archived data that is rarely used

[ ] **C** Use of commodity servers that are used in high volume steaming applications

[X] **D** Use of multiple servers to spread processing and improve performance

You are correct! Scale-out is the use of standby servers that we can switch over to in case of primary server failure. It is also the use of multiple servers to help spread the workload. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

17. SAP HANA security services include which of the following?

Choose the correct answers.

[X] **A** Audit logging

[X] **B** Authentication

[X] **C** Encryption

[ ] **D** Antivirus Protection

[X] **E** Authorization

You are correct! SAP HANA does not handle antivirus protection. Refer to HA100, Unit 2 Architecture of SAP HANA for details.

UNIT 3 Analytical Processing with SAP HANA

UNIT OBJECTIVES

- Understand SAP HANA core modeling
- Describe SAP HANA Live
- Describe Core Data Services (CDS)
- Define SAP HANA spatial processing
- Define text analytics
- Describe predictive analysis in SAP HANA
- Describe graph processing with SAP HANA

Unit 3
Lesson 1

Understanding Core Modeling with SAP HANA

LESSON OVERVIEW
In this lesson we will cover modeling with SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAP HANA core modeling

Modeling in SAP HANA

This lesson explains why modeling in SAP HANA is an important activity that ensures you are using the SAP HANA platform to its full potential.

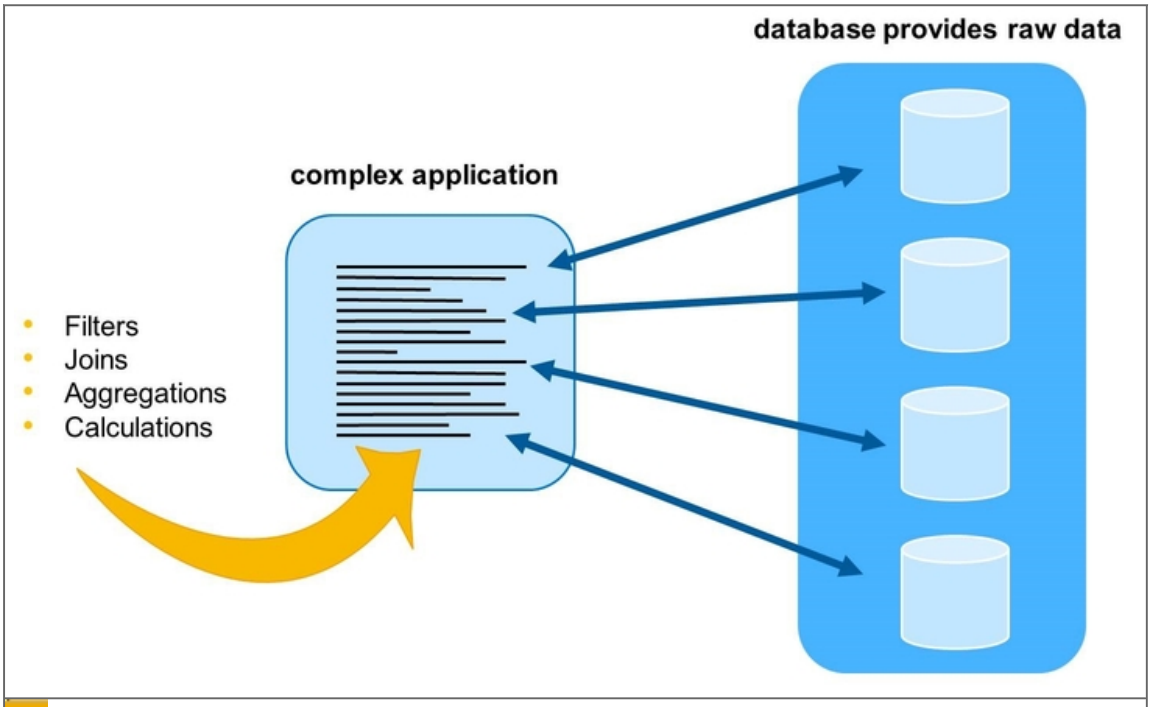


Figure 56: Modeling in the Application

In a traditional application, the role of the database is to simply provide raw data. There is usually very little, or even no data processing in the database.

The raw data is sent from the database directly to the application. The application then begins to process the data by combining it, aggregating it, and performing calculations in order to generate something meaningful.

We can find ourselves moving a lot of raw data between the database and the application. When we move raw data to the application layer we make the application code very complex. This is because It has to deal with the data processing and modeling tasks as well as

managing all of the other process flow control, business logic, User Interface (UI) operations, integrating data from multiple sources, and so on.

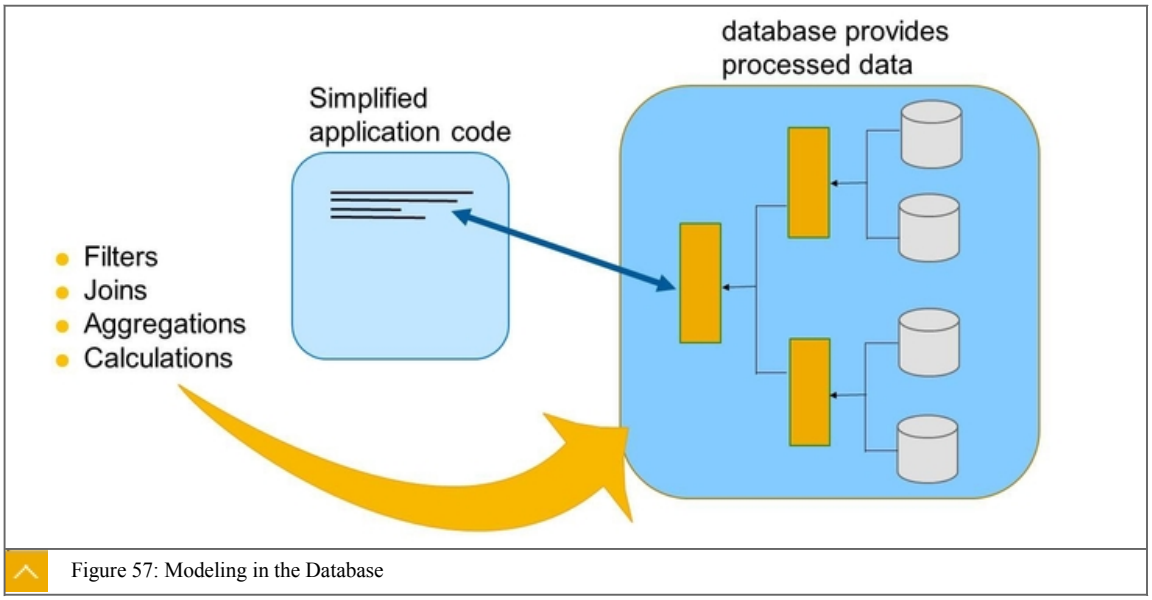


Figure 57: Modeling in the Database

With SAP HANA, we can build a sophisticated modeling layer on top of the database tables. This means we can first process the raw data and turn it into something meaningful in the database before passing it on to the application.

With SAP HANA, we build calculation views to combine data from multiple tables and apply filters, conditions, calculations, and aggregations. The calculation views are developed in SAP HANA using easy-to-use modeling tools, and are stored in SAP HANA alongside the database tables in the database.

Therefore, instead of the application processing the raw data, the application calls the required calculation views and the processing is pushed down to SAP HANA. This is efficient in the following ways:

- The application code is simplified, as it does not have to deal with many data processing tasks. These tasks are pushed down to SAP HANA where in-memory processing takes place.
- The processing on the data is carried out where the data resides, so we do not have to move raw data from the database to the application. We only move the results of the data processing to the application.
- The calculation views can be reused in multiple applications so we avoid redundancy.

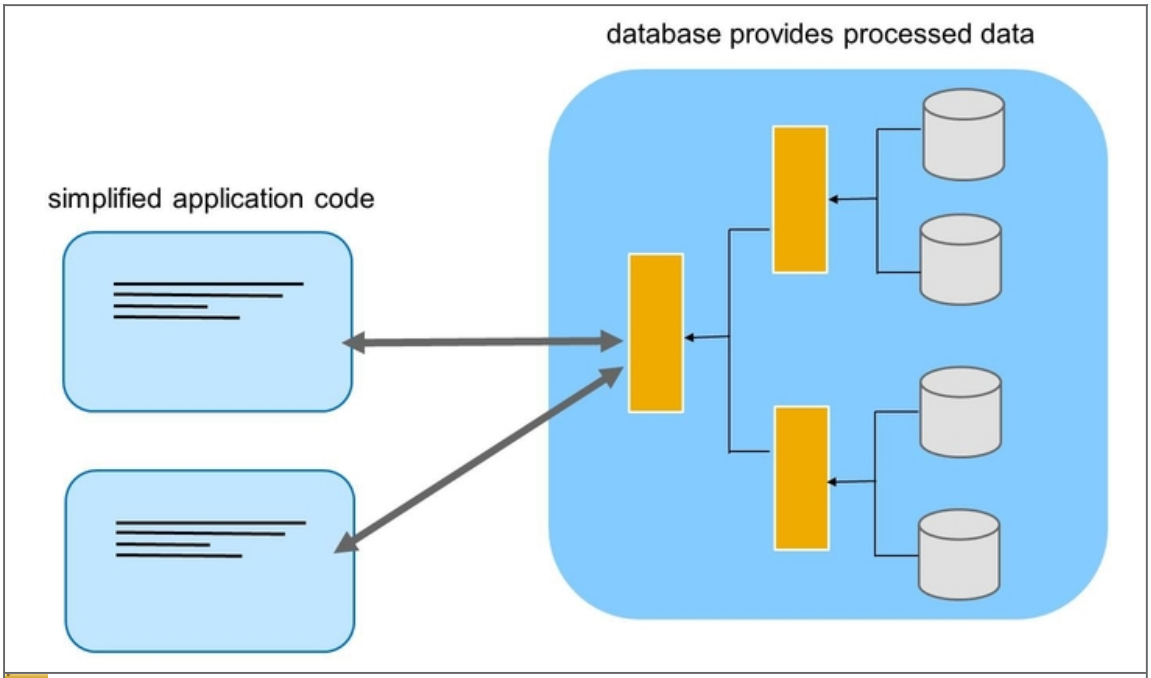


Figure 58: Reuse of Models

In traditional applications, there is a high degree of redundancy in the application code. Developers find themselves continually creating the same code to process data.

When dealing with highly normalized database models, such as those used with SAP Business Suite, there can be many individual tables that need to be called and combined with joins. These joins can often be pushed down to most databases. However, SAP HANA goes beyond helping with just the table joins. SAP HANA can take on the work that was done by the application. SAP HANA takes care of complex calculations and data flow logic, including executing aggregations and disaggregation.

Therefore, you can create an SAP HANA calculation views once and then reuse it. SAP HANA calculation views can contain dynamic placeholders. This means that the applications can pass variables down to the calculation views a response to a filter value that came from a business user. Many of the calculation views can also call procedures that have input parameters.

Calculation views can consume other calculation views. This encourages a high degree of modularization and reuse.

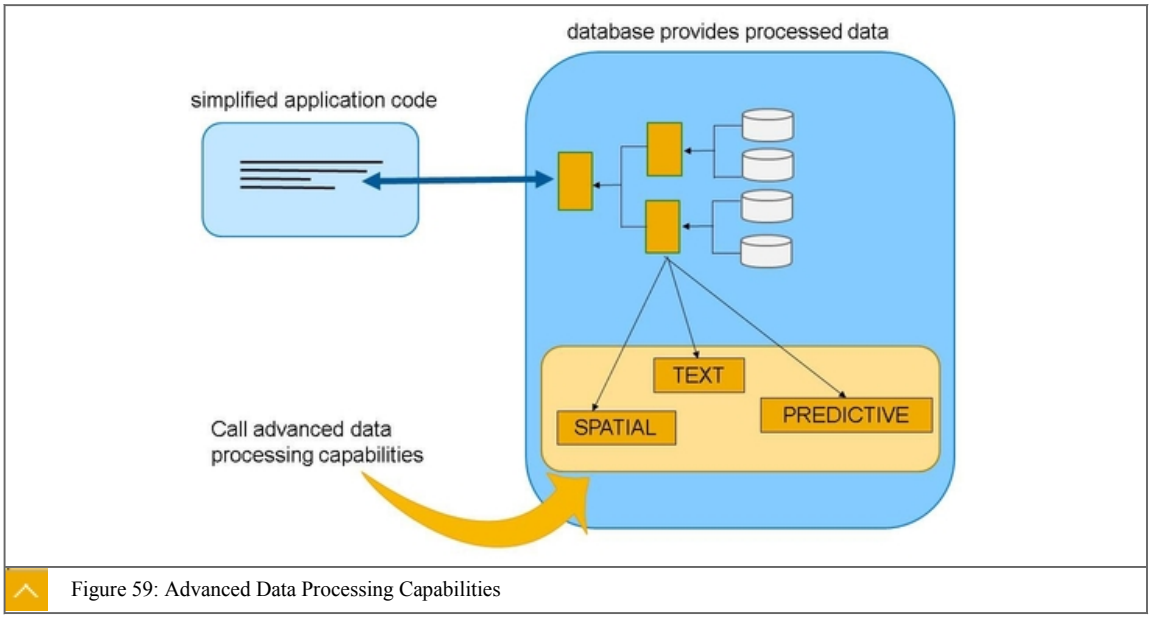


Figure 59: Advanced Data Processing Capabilities

SAP HANA has built-in, advanced data processing capabilities. These include textual, spatial, and predictive functions. Calculation views can easily call these native SAP HANA functions, so applications can leave all the complex processing to SAP HANA.

Core Modeling Versus Advanced Modeling

The term, core modeling (sometimes called view modeling), refers to the development of models that handle common analytical functions. These functions include filtering, aggregation, calculations, and so on.

When we develop models to handle advanced analytical scenarios, such as predictive, spatial, textual, and graph, we refer to this type of modeling as advanced modeling. For now we will focus on core modeling.

Core modeling in SAP HANA begins with the creation of a **calculation view** .

Note:
In earlier releases of SAP HANA, calculations views were part of a family called Information Views. There were originally three members of that family; Attribute View, Analytic View, and Calculation View.

Each of these view types had its own unique features, and typically, all three view types were required. However, since the calculation view has inherited all the features of the two other views, we no longer develop attribute or analytic views. In fact these types of views can be migrated to calculation views using the supplied tools. The calculation view can now do it all, which means we no longer have to be concerned about which view type to use.

Calculation View Creation

When you create a calculation view, you choose various combinations of settings. These settings define four basic types of calculation view.

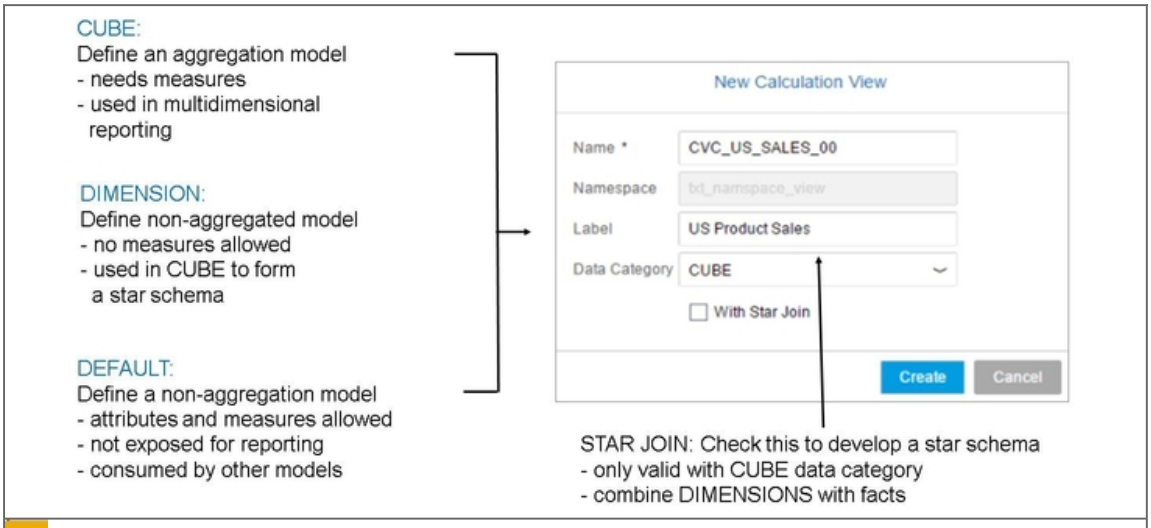
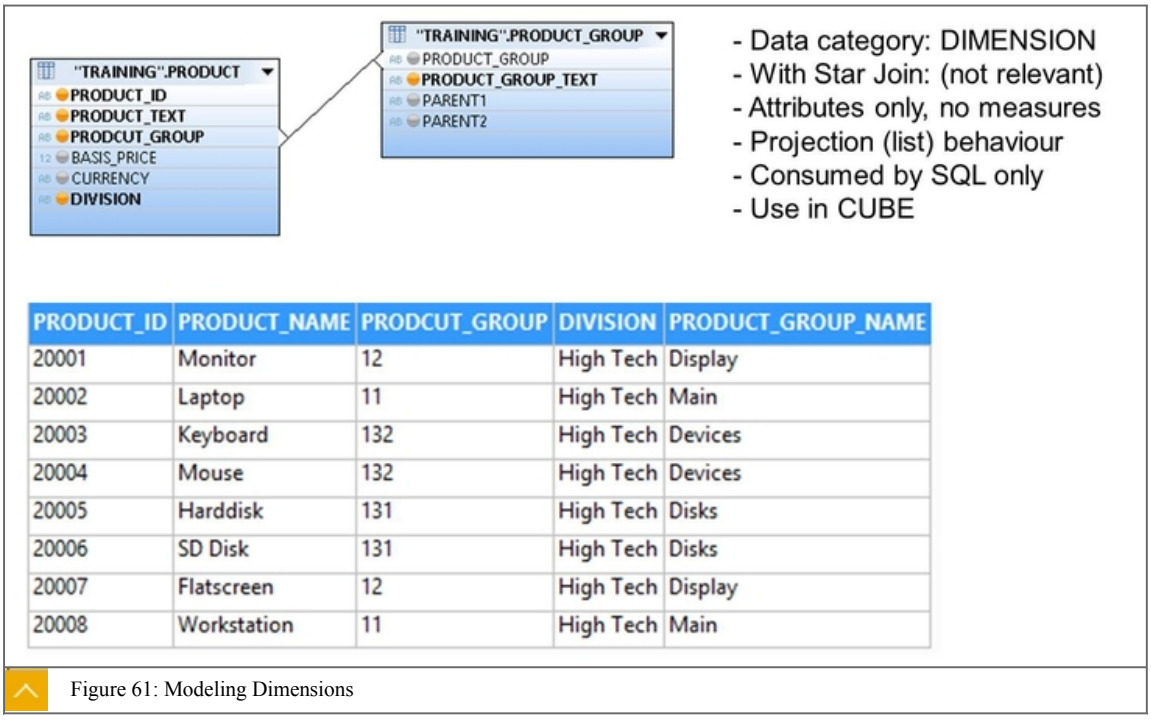


Figure 60: Choosing the Correct Calculation View Settings

The settings are chosen when you first create the calculation view. The four types are as follows:

- Dimension
- Cube without star schema
- Cube with star schema
- Default



PRODUCT_ID	PRODUCT_NAME	PRODCUT_GROUP	DIVISION	PRODUCT_GROUP_NAME
20001	Monitor	12	High Tech	Display
20002	Laptop	11	High Tech	Main
20003	Keyboard	132	High Tech	Devices
20004	Mouse	132	High Tech	Devices
20005	Harddisk	131	High Tech	Disks
20006	SD Disk	131	High Tech	Disks
20007	Flatscreen	12	High Tech	Display
20008	Workstation	11	High Tech	Main

Figure 61: Modeling Dimensions

Dimensions are most likely to be created first. The purpose of a dimension type of calculation view is to define a list of related attributes, such as material, material color, weight, and price. This list can be directly consumed by an application using SQL. However, it is most likely to be found as a component in another calculation view of the type Cube when creating star schemas.

Dimension type calculation views cannot contain measures. They can only contain attributes. Without measures, aggregation is not possible. Reporting tools cannot directly access calculation views of type DIMENSION. Only direct SQL access is allowed.

It can be helpful to think of calculation views of type dimension as master data views. You would not model transaction data using dimension calculation views as no measures can be defined, and measures are for modeling with transactional data.

Note:
Be careful not to confuse measures with attributes that are of a numerical data type, such as integer or decimal. A numeric field can be included in this dimension calculation view but it cannot be modeled as a measure (it must be modeled only as an attribute). This means that there is no aggregation behavior possible. For example, you could include weight but you cannot sum this. The output will appear as a list of all weights.

Data Category — Dimension

To get started with calculation views of type DIMENSION, you need to set the data category to DIMENSION.

You then proceed to define the source tables, the joins, the filters, and the columns that are to be exposed. It is also possible to define additional derived attributes. An example of this could be a new column to generate a weight category based on a range of weights, using an `IF` expression.

You are then able to rename any columns to be more meaningful. Remember that the column names originate from the database tables, and these names can often be meaningless to developers and business users.

YEAR	MONTH	CUSTOMER_ID_1	QUANTITY_SUM	AMOUNT_SUM
2011	12	1000	25	4,520.7
		1001	30	8,262
2012	01	3000	50	11,658
		3001	15	5,255

Figure 62: Modeling Cubes

The next type of calculation view is the type Cube . This type is used to define a dataset comprised of attributes and measures that can be used in a flexible slice and dice format. This

is not a star schema as there are no dimensions defined, but simply a dataset based on one or more transaction tables. These tables can be queried using any combination of the attributes and measures that they include, to create either a line level or an aggregated dataset.

Reporting tools can directly access this type of calculation view. They can also be accessed via SQL.

Data Category — Cube

To get started, set the data category to CUBE.

Do not select the Star Join flag. This is used later, in the next calculation view type.

You then select the table, or tables, which are to be included in the model. Typically, you choose a transaction table so that you have columns from which you can define attributes and measures. It is possible to include more than one table. For example, you may need to include a header and a line item table to form the complete picture of a sales transaction. In this case, you simply join the tables using a JOIN node. You can also merge transaction tables using a UNION node.

Then, select the columns from the tables that are to be exposed. You can optionally set filters and define additional calculated columns.

Then, rename any columns to provide meaningful names to the developer and business user.

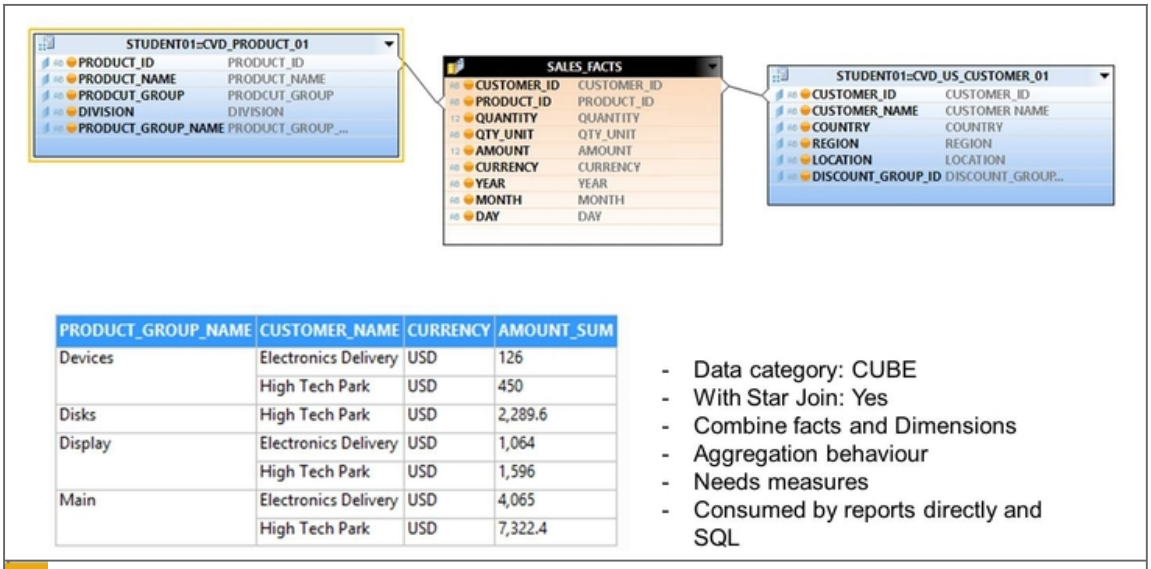


Figure 63: Modeling Star Schemas

The next type of calculation view is the Cube type, but with an additional setting — star schema. This is again based on the cube type of calculation view, but with one or more dimension calculation views joined to the model.

Adding the dimension views enables you to request aggregations of any measures in the fact table by any combination of attributes. You are not limited to just those attributes from the fact table, but also attributes from any dimensions. This significantly increases the analysis possibilities.

This type of calculation view follows the same rules as the cube type. It is consumed directly by reporting tools as well as SQL, and it can include attributes and measures. It is used to present aggregated views of the dataset in the most efficient way.

Data Category — Cube Star Join

To get started, make sure that you set the data category to CUBE and select the Star Join flag.

Select the transaction tables and create joins to combine the transaction tables. Then, choose the columns to expose, set any filters, and create any calculated columns. What you are doing up to this point is forming a fact table that will be used as the hub of the star schema.

The next step is to define the star schema by linking relevant calculation views of type DIMENSION to the fact table.

Then, improve the names of any columns by using the rename function in the Semantic node.

General Purpose Views

The final type of calculation view has the default data category. This type of view is simple and is not meant for multidimensional modeling. It produces flattened result sets that can contain attributes and measures. It is not visible to reporting tools.

These types of views are usually used as data sources to cube type views. This type of view does not expose its metadata to the consuming applications as the other views do. It can help to think of this type of view as an internal view.

SAP HANA Modeling Interfaces

Calculation views can be created in the SAP HANA Studio and also in the Web IDE for SAP HANA. However, a calculation view that is created in Studio is not visible in Web IDE (and vice-versa). The reason for this is that each interface client uses a completely different repository framework for the storage of their views and these are not interchangeable. Studio is used for classic XS development and Web IDE is used for XSA development. It is possible to develop models in both environments in the same SAP HANA system, but SAP strongly recommends creating all new modeling artifacts in the newer and more powerful XSA framework. Many customers will be moving from XS to XSA development and during the transition will need to operate both interfaces.

When it comes to developing calculation views, the sequence of steps is similar between the two interfaces and they both produce the same activated calculation view which is deployed as a column view in the database. If you learn modeling in Studio, it is very easy to move your skills to Web IDE.

SAP HANA Studio — Modeler Perspective

With SAP HANA Studio, once you open the Modeler perspective, you can create and maintain all types of modeling objects including those that are now deprecated (e,g. attribute and analytic views). This is helpful if you need to access these older types of objects. Web IDE cannot provide access to attribute or analytic views and only supports the calculation view (because that is the only type you should be building).

With SAP HANA Studio Modeler perspective, you can create the following XS based core modelling objects:

- Attribute View (deprecated)
- Analytic View (deprecated)
- Calculation View
- Decision Table
- Procedure
- Analytic Privilege

You will need to use the Development perspective to create these:

- Functions
- Flowgraph
- CDS artifacts

Beyond modeling, SAP HANA Studio also offers many other useful functions and tools that a modeler may find helpful. These functions include a flat file import wizard, SLT cockpit, and a helpful Quick View pane where many modeler tools and utilities are found in a convenient list. So, even for developers and modelers who use Web IDE, it is possible that they may return to Studio to use some of the tools that are not yet available in Web IDE.

Web IDE for SAP HANA, Modelling Objects

With Web IDE, you can create the following core modelling objects:

- Calculation View
- Procedure
- Function
- Flowgraph
- Analytic Privilege
- CDS artifacts

For developers and modelers, moving from the Studio to Web IDE is straightforward and requires no re-training. All the familiar modeling features are used.

Unit 3
Exercise 6

Create Dimension Calculation Views

Exercise Objectives

In this exercise, you learn how to create a calculation view of the type dimension. You will later join this view to the sales fact data in another calculation view of the type cube with star join.

Business Example

You need to make available the following two calculation views in order to provide the master data attributes that are needed for inclusion in the calculation view of type cube with star join:

- A calculation view of type dimension, to expose information relating only to US customers
- A calculation view of type dimension, to expose product master data

Before starting this exercise, we advise that you preview the content for each of the required tables in order to familiarize yourself with the data and also think about the joins that might be used:

Refer to the course section that covers the Web IDE Database Explorer to learn how to access and browse the tables.

Database Schema: TRAINING

Column Tables: CUSTOMER, PRODUCT, and PRODUCT_GROUP.

Note:
In this exercise, replace ## with your own student number.

Task 1: Create a calculation view of type dimension for customer master data attributes

1. Start the Web IDE and open the Development view.
2. Create a new folder with the name **`exercises`** as a sub-folder under HA100 → HDB → src .
3. In your exercises sub–folder, create a new calculation view of the type dimension, with the properties provided below:

   Table 8: Calculation View Properties

Field	Value
Name	**`CVD_US_CUSTOMER_##`**
Label	**`US Customers`**
Data Category	Dimension

4. Create a new Projection node and add the CUSTOMER table, from the TRAINING schema, to it.
5. Add all fields to the output.
6. To show entries only for the US (United States), filter the COUNTRY table column.
7. Connect the Projection nodes.
8. Add all columns to the output of the final Projection node.
9. Change CUSTOMER_TEXT to CUSTOMER_NAME for the **name** of the column ('name' is the technical name – no spaces allowed).
10. Change CUSTOMER_TEXT to Name of customer for the **label** of the column ('label ' is the user-friendly name).
11. Save and build the new calculation view checking the console to ensure that the calculation view has been successfully built.
12. Preview the data for CVD_US_CUSTOMER_## . Check that you only see US customers and that all columns from the source table are shown.

Task 2: Create a calculation view of type dimension for product master data attributes

1. In your exercises subfolder, create a new calculation view of the type dimension with the properties provided below:

Table 9: Calculation View Properties 2

Field	Value
Name	**CVD_PRODUCTS_##**
Label	**All Products**
Data Category	Dimension

2. Add the PRODUCT and PRODUCT_GROUP tables from the TRAINING schema to a new join node.
3. Join the column PRODUCT_GROUP, from the table PRODUCT, to the column PRODUCT_GROUP, from the table, PRODUCT_GROUP, using an inner join and n..1 cardinality.
4. From the HA100 :: PRODUCT table, map only the four fields PRODUCT_ID, PRODUCT_TEXT, PRODUCT_GROUP, and DIVISION to the output.
5. From the HA100 :: PRODUCT_GROUP table, map only the field PRODUCT_GROUP_TEXT to the output.
6. Connect the join node to the projection node.
7. Add all columns to the output of the final projection node.
8. Change PRODUCT_TEXT to PRODUCT_NAME for both the label and also the name
9. Save and build the new calculation view. Then check the console to ensure that the calculation view has been successfully built.

10. Preview the data for the calculation view CVD_PRODUCT_## . Check that you see the product group text displayed against all products (this proves that the join is working).

Unit 3
Solution 6

Create Dimension Calculation Views

Exercise Objectives

In this exercise, you learn how to create a calculation view of the type dimension. You will later join this view to the sales fact data in another calculation view of the type cube with star join.

Business Example

You need to make available the following two calculation views in order to provide the master data attributes that are needed for inclusion in the calculation view of type cube with star join:

- A calculation view of type dimension, to expose information relating only to US customers
- A calculation view of type dimension, to expose product master data

Before starting this exercise, we advise that you preview the content for each of the required tables in order to familiarize yourself with the data and also think about the joins that might be used:

Refer to the course section that covers the Web IDE Database Explorer to learn how to access and browse the tables.

Database Schema: TRAINING

Column Tables: CUSTOMER, PRODUCT, and PRODUCT_GROUP.

Note:
In this exercise, replace ## with your own student number.

Task 1: Create a calculation view of type dimension for customer master data attributes

1. Start the Web IDE and open the Development view.
   a) On the left of the screen you should see three buttons arranged vertically. Select the top button, which is the Development view.
2. Create a new folder with the name **`exercises`** as a sub-folder under HA100 → HDB → src .
   a) In your workspace expand the folders HA100 → HDB → src .
   b) Right click the src folder and choose New → Folder .
   c) Enter the folder name **`exercises`** and choose OK.
3. In your exercises sub–folder, create a new calculation view of the type dimension, with the properties provided below:

Table 8: Calculation View Properties

Field	Value
Name	**CVD_US_CUSTOMER_##**
Label	**US Customers**
Data Category	Dimension

a) In the Development view, right-click the exercises subfolder and choose New → Calculation View .

b) Enter the properties from the table, Calculation View Properties .

c) Choose Create .

4. Create a new Projection node and add the CUSTOMER table, from the TRAINING schema, to it.

a) From the node selection toolbar, click on the Projection node then click in the empty space at the bottom of the dataflow to add the new node.

The new Projection node should appear directly below the existing Projection node.

b) Make sure the newly added Projection node is selected and open the data source selector, by choosing the + icon that appears on the right of the node.

c) In the search field, enter **Cust**.

A list of data sources that match your search will appear.

d) Highlight the row where the table CUSTOMER appears.

e) Choose Finish .

5. Add all fields to the output.

a) Choose the Projection node that you just added. Ensure that you are displaying the Mapping tab and then right-click the table header HA100 :: CUSTOMER , which is found under Data Sources .

b) In the context menu of the table header, choose Add to Output .

All fields should now appear on the right, under Output Columns .

6. To show entries only for the US (United States), filter the COUNTRY table column.

a) Choose the Filter Expression tab.

b) In the Elements pane, expand the Columns .

c) Double-click Country so it appears in the expression canvas.

d) From the Operators pane, choose **=**.

e) Enter **'** (a single quote) and a second, closing single quote automatically appears. Between the quotes, in uppercase, enter **US**.

7. Connect the Projection nodes.

a) Select the Projection node that you added so that the action symbols appear to the right.

b) Drag a line between the arrow symbol and the circle at the bottom of the Projection node above.

The Projection nodes are now connected.

8. Add all columns to the output of the final Projection node.

a) Select the upper-most Projection node which is directly below the Semantics node. Ensure that you are displaying the Mapping tab and then right-click the header Projection_1 .

b) In the context menu, choose Add to Output .

All fields should appear on the right, under Output Columns .

9. Change CUSTOMER_TEXT to CUSTOMER_NAME for the **name** of the column ('name' is the technical name – no spaces allowed).

a) In the Semantic node, click inside the CUSTOMER_TEXT field, under the Name column and enter **`CUSTOMER_NAME`**.

10. Change CUSTOMER_TEXT to Name of customer for the **label** of the column ('label ' is the user-friendly name).

a) In the Semantic node, click inside the CUSTOMER_TEXT field, under the Label column. Enter **`Name of customer`**.

11. Save and build the new calculation view checking the console to ensure that the calculation view has been successfully built.

a) Choose the Save button, or use the menu path File → Save .

b) Choose the menu path Build → Build Selected Files .

In the console, which appears at the bottom right of the screen, you will see a log. Ensure that you can see that the build completed successfully.

12. Preview the data for CVD_US_CUSTOMER_## . Check that you only see US customers and that all columns from the source table are shown.

a) Right-click your calculation view, which appears on the Development tree on the left of the screen. In the context menu, choose Data Preview .

b) Check the Country column to ensure that you only see US customers and all six columns from the original database table are displayed.

Task 2: Create a calculation view of type dimension for product master data attributes

1. In your exercises subfolder, create a new calculation view of the type dimension with the properties provided below:

Table 9: Calculation View Properties 2

Field	Value
Name	**`CVD_PRODUCTS_##`**
Label	**`All Products`**
Data Category	Dimension

a) In the Development view, right-click the exercises subfolder and choose New → Calculation View .

b) Enter the properties provided in the table, Calculation View Properties 2 .

c) Choose Create .

2. Add the PRODUCT and PRODUCT_GROUP tables from the TRAINING schema to a new join node.

a) From the node selection toolbar, choose the join node then choose the empty space at the bottom of the data flow to add the new node.

The new node should appear directly below the existing projection node.

b) Choose the newly added join node. To open the data source selector, choose the + symbol that appears on the right.

c) In the Search field, enter **`Product`**.

d) Highlight the rows where the tables PRODUCT and PRODUCT_GROUP appear.

e) Choose Finish .

The two tables should appear in the canvas side-by-side.

Note:
If the columns of the tables do not appear on the canvas, simply select a different node then return to the join node. The columns should now appear (this is a Web IDE refresh issue that will be fixed).

3. Join the column PRODUCT_GROUP , from the table PRODUCT , to the column PRODUCT_GROUP , from the table, PRODUCT_GROUP , using an inner join and n..1 cardinality.

a) On the canvas, choose the create join icon in the toolbar located in the top right. The Join Creation editor appears.

b) Drag the column PRODUCT_GROUP from the left table (PRODUCT) to the PRODUCT_GROUP column of the right table (PRODUCT_GROUP).

c) Expand General Properties and from the Join Type selector choose Inner .

d) In the Cardinality selector choose n..1 .

e) Choose OK.

4. From the HA100 :: PRODUCT table, map only the four fields PRODUCT_ID, PRODUCT_TEXT, PRODUCT_GROUP , and DIVISION to the output.

a) Make sure that the join node that you recently added is still highlighted and choose the Mapping tab.

b) Under the header HA100 :: PRODUCT , you should see a list of all available source fields from this table. Right-click only the column PRODUCT_ID and in the context menu, choose Add to Output .

The single column appears on the right side, under the Output Columns pane.

c) Repeat substep b for PRODUCT_TEXT, PRODUCT_GROUP , and DIVISION.

5. From the HA100 :: PRODUCT_GROUP table, map only the field PRODUCT_GROUP_TEXT to the output.

a) Under the header HA100 :: PRODUCT_GROUP you should see a list of all available source fields. Right-click PRODUCT_GROUP_TEXT and in the context menu, choose Add to Output .

The field appears on the right side, under the Output Columns pane. You should now have five output fields.

6. Connect the join node to the projection node.

a) Choose the join node (the one you created) so that the action symbols appear to the right.

b) Drag a line between the arrow symbol and the circle at the bottom of the projection node above.

The nodes are now connected.

7. Add all columns to the output of the final projection node.

a) Choose the projection node. Ensure that you are displaying the Mapping tab and right-click the header Join_1 under Data Sources .

b) In the context menu, choose Add to Output .

All five fields should appear on the right side, under the Output Columns pane.

8. Change PRODUCT_TEXT to PRODUCT_NAME for both the label and also the name

a) In the Semantic node, click inside the PRODUCT_TEXT field, under the Name column. Enter **PRODUCT_NAME**.

b) Click inside the field PRODUCT_TEXT, under the Label column. Enter **PRODUCT_NAME**.

9. Save and build the new calculation view. Then check the console to ensure that the calculation view has been successfully built.

a) Choose Save , or use the menu path File → Save .

b) Choose Build → Build Selected Files .

In the console, which appears at the bottom of the screen, you should see a log. Ensure that you can see that the build completed successfully.

10. Preview the data for the calculation view CVD_PRODUCT_## . Check that you see the product group text displayed against all products (this proves that the join is working).

a) Right-click your calculation view that appears on the Development tree on the left of the screen. In the context menu, choose Data Preview .

b) Check that the PRODUCT_GROUP_TEXT column appears for each product and that the five output columns you chose are displayed.

Unit 3
Exercise 7

Create a Calculation View of type Cube with Star Join

Exercise Objective

In this exercise, you learn how to build a calculation view of type cube with star join that combines the two dimension views (that you created earlier) with sales transactions to form a multi-dimensional model for flexible analysis.

Business Example

You have been asked to create a model that enables users to thoroughly explore product sales data for US customers. It should be possible to present all sales figures, aggregated by any combination of customer and product attributes.

Before you start, preview the sales fact table to familiarize yourself with the data.

Refer to the course section, which covers the Database Explorer to learn how to access and browse the tables.

Database Schema: TRAINING

Column Table: SALES_DATA.

Columns of interest: CUSTOMER_ID, PRODUCT_ID, QUANTITY, QTY_UNIT, AMOUNT, CURRENCY, and SQL_DATE.

Note:
In this exercise, replace ## with your own, two-digit student number.

Task 1: Create the basic calculation view of type cube with star join

1. In the exercises folder, create a new calculation view with the settings provided below:

   Table 10: Calculation View Properties

Field	Value
Name	`CVC_US_SALES_##`
Label	`US Product Sales`
Data Category	CUBE
With Star Join	Select

2. Add an aggregate node to the flow and include the table SALES_DATA, from the TRAINING schema, to this node.

3. A star schema needs a central fact table. To create this, add the seven columns from the table, Column Names , to the output.

   Table 11: Column Names

Column Name
CUSTOMER_ID
PRODUCT_ID
QUANTITY
QTY_UNIT
AMOUNT
CURRENCY
SQL_DATE

4. Change the name of the aggregation node from Aggregation_1 to SALES_FACTS to provide a better meaning for the fact data.
5. Connect the aggregation node to the star join node.
6. To create your star schema, add the CVD_US_CUSTOMER_## and CVD_PRODUCT_## dimension views to the star join node.
7. Join the SALES_FACTS data source to the two dimension views.
8. From the star join node, map the five columns, QUANTITY, QTY_UNIT, AMOUNT, CURRENCY, and SQL_DATE to the output.
9. Change AMOUNT to SALES_REVENUE and SQL_DATE to SALES_DATE.
10. In the semantic node, check that the correct column type has automatically been assigned to the columns SALES_REVENUE and QUANTITY.

    They should both be defined as a Measure and all the other columns should be defined as an Attribute .
11. Save and build the new calculation view. Then, check the console to ensure that the calculation view has been successfully built.
12. Preview the raw tabular data for CVC_US_SALES_## . Check that you see the measures from the fact table are supported by attributes from both dimensions.
13. Use the Analysis function to display the total sales quantity by product group.

Task 2: Enhance your calculation view with a calculated column

1. To indicate whether delivery is free or chargeable based on the customer location, add a calculated column to the calculation view based on a column engine expression. Use the information provided below:

   Table 12: Calculated Column Data

Section	Field	Value
General	Name	`Delivery`

Section	Field	Value
General	Label	**Delivery Charge**
General	Data Type	**VARCHAR**
General	Length	**10**
Semantics	Column Type	**Attribute**
Expression	Formula	`If ("LOCATION"='Philadel-phia','FOC','CHARGEABLE')`

2. Save and build the calculation view. Then check the console log to ensure that the calculation view has been successfully built.

3. Preview the data for CVC_US_SALES_## . Check that you see the measures from the fact table supported by attributes from the dimensions.

Unit 3
Solution 7

Create a Calculation View of type Cube with Star Join

Exercise Objective

In this exercise, you learn how to build a calculation view of type cube with star join that combines the two dimension views (that you created earlier) with sales transactions to form a multi-dimensional model for flexible analysis.

Business Example

You have been asked to create a model that enables users to thoroughly explore product sales data for US customers. It should be possible to present all sales figures, aggregated by any combination of customer and product attributes.

Before you start, preview the sales fact table to familiarize yourself with the data.

Refer to the course section, which covers the Database Explorer to learn how to access and browse the tables.

Database Schema: TRAINING

Column Table: SALES_DATA.

Columns of interest: CUSTOMER_ID, PRODUCT_ID, QUANTITY, QTY_UNIT, AMOUNT, CURRENCY, and SQL_DATE.

Note:
In this exercise, replace ## with your own, two-digit student number.

Task 1: Create the basic calculation view of type cube with star join

1. In the exercises folder, create a new calculation view with the settings provided below:

Table 10: Calculation View Properties

Field	Value
Name	**CVC_US_SALES_##**
Label	**US Product Sales**
Data Category	CUBE
With Star Join	Select

a) In the content tree of the Development view, right-click the exercises folder and choose New → Calculation View .

b) Enter the properties provided in the table, Calculation View Properties .

c) Choose Create .

2. Add an aggregate node to the flow and include the table SALES_DATA, from the TRAINING schema, to this node.

a) From the node selection toolbar, click on the Create Aggregation icon and then click in the empty space at the bottom of the data flow to add the new node.

The new aggregate node should appear directly below the existing star join node.

b) Choose the newly-added aggregate node and open the data source selector by choosing the + symbol.

c) In the Search field, enter **`sales`**.

d) Highlight the row where the table SALES_DATA appears.

e) Choose Finish .

3. A star schema needs a central fact table. To create this, add the seven columns from the table, Column Names , to the output.

Table 11: Column Names

Column Name
CUSTOMER_ID
PRODUCT_ID
QUANTITY
QTY_UNIT
AMOUNT
CURRENCY
SQL_DATE

a) Make sure that the aggregate node is still highlighted and choose the Mapping tab.

b) Under the header HA100 :: SALES_DATA , you should see a list of all available source fields. Right-click each field from the list provided above, and in the context menu, choose Add to Output .

The seven selected fields appear on the right under Output Columns .

4. Change the name of the aggregation node from Aggregation_1 to SALES_FACTS to provide a better meaning for the fact data.

a) Carefully click on the label of the node Aggregate_1 and overwrite it with the new label **`SALES_FACTS`**.

5. Connect the aggregation node to the star join node.

a) Select the aggregation node so that the action symbols appear to the right.

b) Drag a line between the arrow symbol and the circle at the bottom of the star join node above.

6. To create your star schema, add the CVD_US_CUSTOMER_## and CVD_PRODUCT_## dimension views to the star join node.
   a) Select the star join node then open the data source selector by choosing the + symbol.
   b) In the Search field, enter **_##** (don't forget to replace ## with your assigned number).
      The two dimension calculation views that you created earlier should appear.

      Note:
      Remember, you can also use the ready-made calculation views of type dimension that end with _00 in case you didn't create one or both of your own from the previous exercise.

   c) Highlight the two rows where the calculation views CVD_US_CUSTOMER_## and CVD_PRODUCT_## appear.
   d) Choose Finish .
      They now appear in the canvas alongside the sales facts.

      Note:
      If the columns of the dimension calculation views do not appear, simply select a different node, then return to the star join node. The columns should now appear (this is a Web IDE refresh issue that will be fixed).

   e) Using drag and drop, arrange the views so that the sales facts are in the center of the canvas with the dimension views at each side. This makes it easier to create a star schema in the next step.
7. Join the SALES_FACTS data source to the two dimension views.
   a) Draw a line from PRODUCT_ID in the SALES_FACTS data source to the PRODUCT_ID field in the CVD_PRODUCT_## dimension view.
   b) Draw a line from CUSTOMER_ID in the SALES_FACTS data source to the CUSTOMER_ID field in the CVD_US_CUSTOMER_## dimension view.
8. From the star join node, map the five columns, QUANTITY, QTY_UNIT, AMOUNT, CURRENCY, and SQL_DATE to the output.
   a) Select the star join node and then select the Mapping tab at the top of the screen.
   b) Under the header SALES_FACTS , you should see a list of all seven available source fields. Right-click QUANTITYand in the context menu, choose Add to Output .
      The field should appear on the right side, under the Output Columns pane.
   c) Repeat sub-step b for QTY_UNIT, AMOUNT, CURRENCY, and SQL_DATE.
9. Change AMOUNT to SALES_REVENUE and SQL_DATE to SALES_DATE.
   a) Choose the semantics node. In the Name and Label columns for AMOUNT, enter **SALES_REVENUE**.
   b) Choose the semantics node. In the Name and Label columns for SQL_DATE, enter **SALES_DATE**.

10. In the semantic node, check that the correct column type has automatically been assigned to the columns SALES_REVENUE and QUANTITY.

They should both be defined as a Measure and all the other columns should be defined as an Attribute .

a) Choose the Semantics node.

b) Verify that the Type column displays the correct type for each row.

11. Save and build the new calculation view. Then, check the console to ensure that the calculation view has been successfully built.

a) Choose Save , or use the menu path File → Save .

b) Choose Build → Build Selected Files .

c) In the console, which appears at the bottom of the screen, you should see a log. Ensure that you can see that the build completed successfully.

12. Preview the raw tabular data for CVC_US_SALES_## . Check that you see the measures from the fact table are supported by attributes from both dimensions.

a) Right-click your calculation view that appears on the Development tree on the left of the screen. In the context menu, choose Data Preview .

b) The Raw Data tab is defaulted and you should now check that the master data attributes appear alongside the sales amounts and quantities.

13. Use the Analysis function to display the total sales quantity by product group.

a) Still in the Data Preview , choose the Analysis tab.

b) From the Attributes , drag PRODUCT_GROUP_TEXT to the Label Axis .

c) From the Measures , drag QUANTITYto the Value Axis .

Task 2: Enhance your calculation view with a calculated column

1. To indicate whether delivery is free or chargeable based on the customer location, add a calculated column to the calculation view based on a column engine expression. Use the information provided below:

Table 12: Calculated Column Data

Section	Field	Value
General	Name	**Delivery**
General	Label	**Delivery Charge**
General	Data Type	**VARCHAR**
General	Length	**10**
Semantics	Column Type	**Attribute**
Expression	Formula	`If ("LOCATION"='Philadel-phia','FOC','CHARGEABLE')`

a) In the calculation view choose the star join node and choose the tab Calculated Columns .

b) Choose the + symbol and in the context menu, choose Calculated Column .

c) Using the first five entries from table, Calculated Column Data , define the basic settings of the calculated column.

d) Choose the Expression Editor button at the bottom of the screen.

e) Change the Language setting to Column Engine .

f) From the Functions pane, double-click the If () function that you will find when you expand the Misc Functions category.

g) In the Expression Editor pane, remove the int placeholder and leave the cursor exactly in its place.

h) In the Elements pane, expand Calculation Views and then expand CVD_US_CUSTOMER_## . Double-click the column Location so that it appears immediately after the first parenthesis and before the first comma.

i) From the Operators pane, choose the = symbol so that it appears immediately after "LOCATION".

j) Immediately after the = symbol, type **`'Philadelphia'`** and be sure to include both single quotes to show that it is a string value.

k) After the first comma remove the arg1 placeholder and replace it by typing **`'FOC'`** remembering to include both single quotes.

l) After the second comma remove the arg2 placeholder and replace it by typing **`'CHARGEABLE'`** remembering to include both single quotes.

The full expression should be
`If("LOCATION"='Philadelphia','FOC','CHARGEABLE')`.

m) To check for errors, choose the Validate Syntax button.

2. Save and build the calculation view. Then check the console log to ensure that the calculation view has been successfully built.

a) Choose Save , or use the menu path File → Save .

b) Choose Build → Build Selected Files .

c) In the console, which appears at the bottom of the screen, you should see a log. Ensure that you can see that the build completed successfully.

3. Preview the data for CVC_US_SALES_## . Check that you see the measures from the fact table supported by attributes from the dimensions.

a) Right-click your calculation view that appears on the Development tree on the left of the screen. In the context menu, choose Data Preview .

b) Check that the Delivery column now shows either 'FOC' or 'CHARGEABLE' for each product depending on whether the customer is based in Philadelphia or not.

Additional Considerations for Core Modeling

Introducing Functions

In the previous concept, we focused on calculation views. Calculation views are built using a graphical approach and no coding is required. However, sometimes the graphical approach does not provide all the functions and data flow logic you require for a complex calculation view. This is when you use functions. Functions are built using SQLScript and offer lots of flexibility to write simple or complex logic.

Note:
There are actually two type of functions — **scalar functions** (return a single value, such as current date) and **table functions** (return a tabular data set of many rows). In this section we focus on table functions.

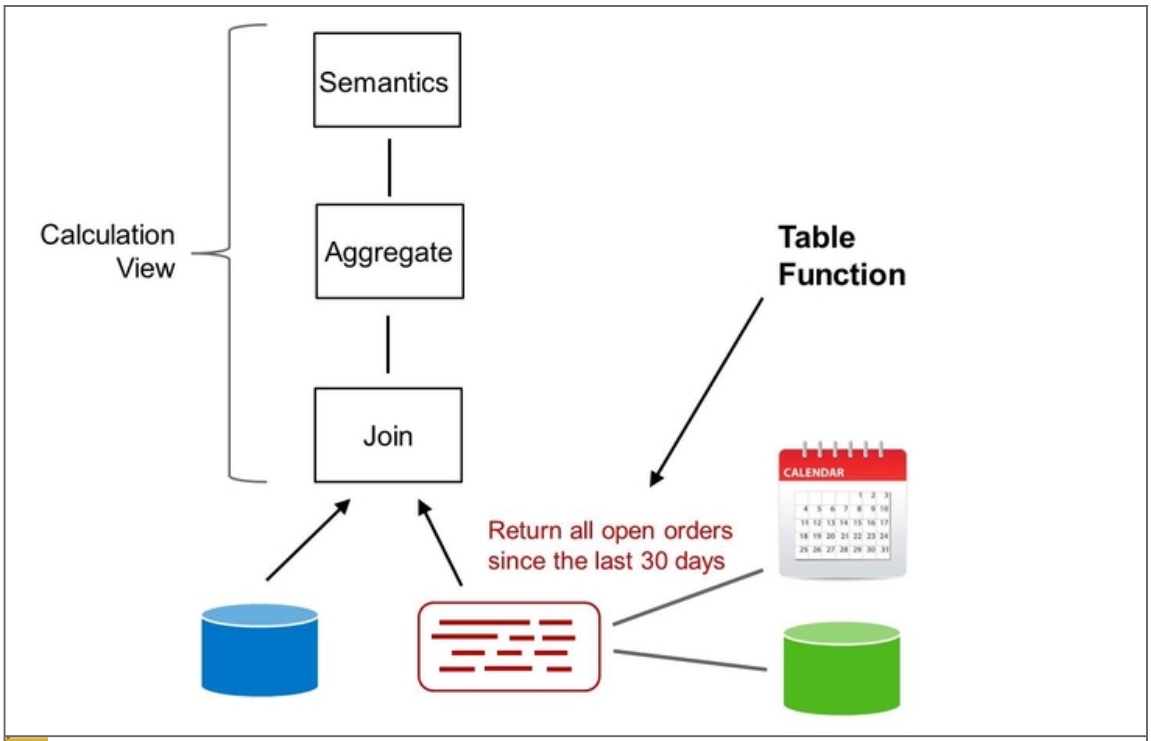


Figure 64: Table Functions

In SAP HANA modeling, table functions are typically used to generate a tabular data set that is used as a data source in a calculation view. Table functions can be used in SQLScript in a FROM clause of a select statement (in other words wherever a standard table is used). A table function encapsulates the logic in a reusable form so that it can be used many times in different artifacts.

Table functions can accept one or more input parameters. Table functions are read-only; that is, they cannot be used to change data. Table functions produce exactly only one tabular output. Table functions can also call other functions.

Introducing Procedures

Procedures define reusable data processing logic that can be used to enhance a calculation view. Procedures are very similar to functions in that they are written in SQL Script and can have one or more inputs and they always have outputs. However, procedures can produce

multiple output data sets of different structures, whereas a table function can only return one tabular output data set. Procedures cannot be used as data sources to calculation views. They are used throughout the calculation view, especially where some processing logic is required, for example, to automatically derive values for an input parameter, or to return values used in analytic privileges.

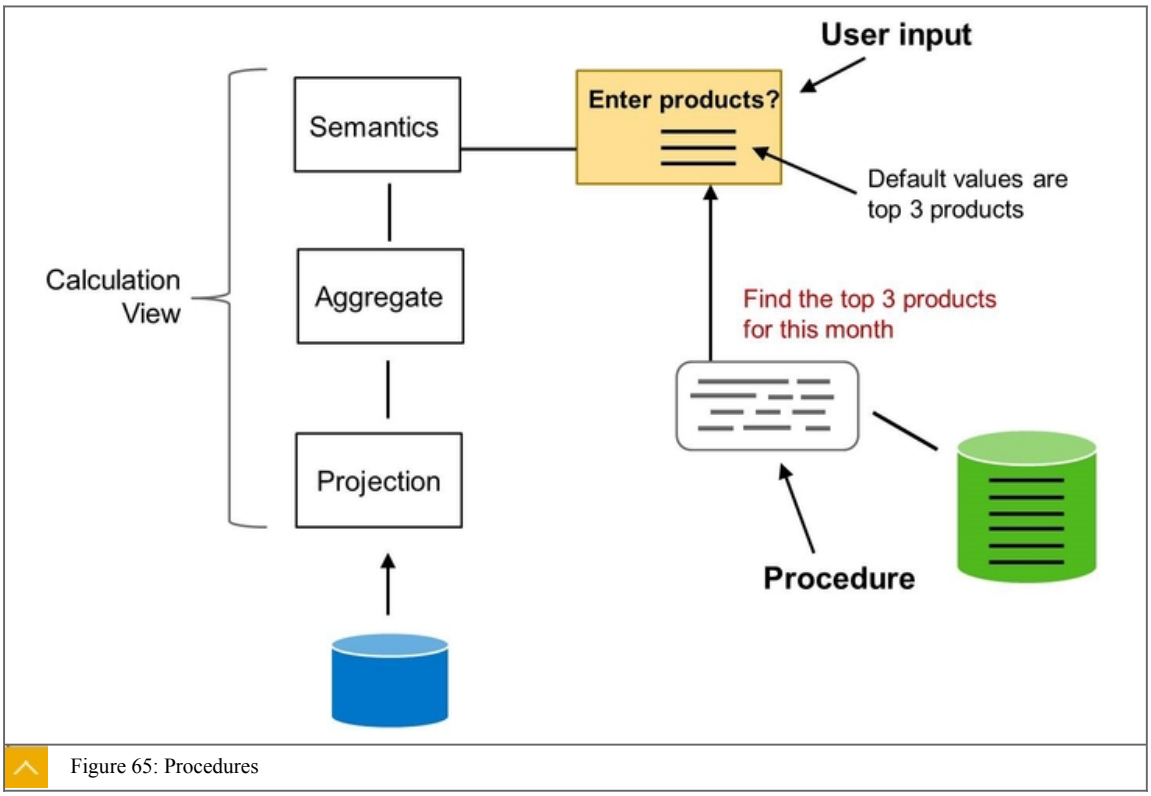


Figure 65: Procedures

A procedure can be called directly from SQLScript which means it can be called from a function or even another procedure.

Procedures used within modeling are mostly used as read only. In that case they are called stateless (or side-effect free), as they don't alter any data in the database. However, procedures can also be used to update, insert, and delete data. These procedures are called stateful, and they are not allowed when called from calculation views. Stateful procedures are more likely to be used by developers building applications.

Security Considerations in Modeling

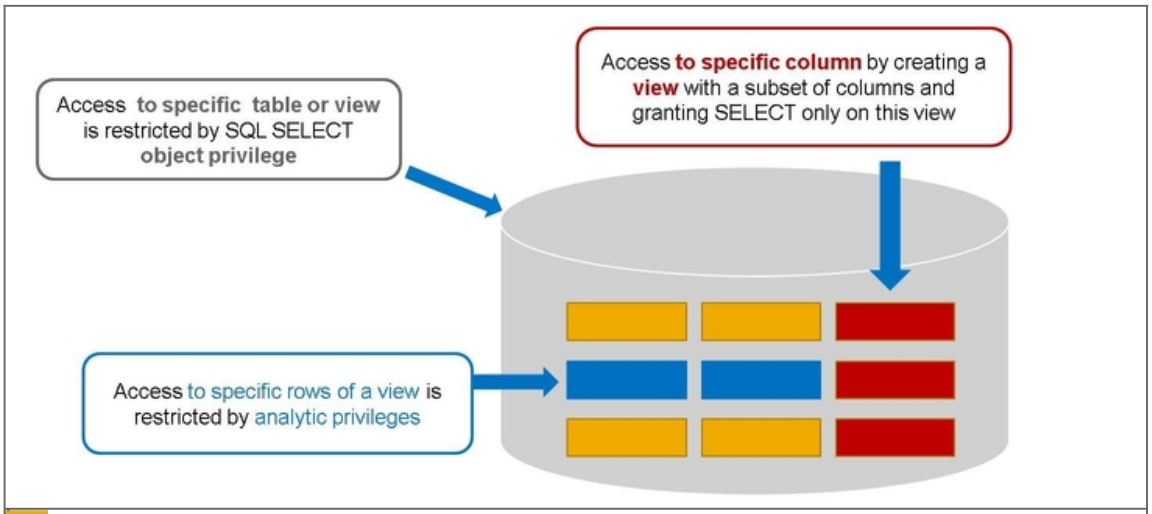


Figure 66: Security Considerations in Modeling

Unless you grant access to users, they will not be able to view any data from your calculation views. There are two levels of security, as follows:

1. The first level of access required is to the actual database objects. In this case we mean the calculation view and the source tables and functions that are included in the calculation view. This is achieved by granting a SELECT privilege on the calculation view. This privilege is granted to the user, or more likely, to the role the user is assigned to.
2. They then need to have access at the row (business data) level. This is achieved by defining an **analytic privilege** and assigning it to the user, or the role to which the user is assigned.

An analytic privilege is an object that is created in the SAP HANA Studio or Web IDE for SAP HANA. It describes for each calculation view the conditions for data access. The conditions can be simple such as User 1 can see company A. The conditions can also be more complex such as User 1 can see company A and B but only between 2017–2018. A user can have multiple analytic privileges. Analytic privilege logic can also be written using SQLScript in procedures.

Note:
For more information about security in SAP HANA, refer to the detailed course, HA240.

LESSON SUMMARY
You should now be able to:

- Understand SAP HANA core modeling

Unit 3
Lesson 2

Understanding Virtual Data Models with SAP HANA

LESSON OVERVIEW
In this lesson we will describe the two types of SAP HANA virtual data models delivered by SAP.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe SAP HANA Live
- Describe Core Data Services (CDS)

SAP HANA Live

SAP HANA Live is a comprehensive set of SAP-supplied calculation views that expose live transactional and master data tables from SAP Business Suite applications. These include SAP ERP, SAP CRM powered by SAP HANA, SAP SCM, and so on. SAP HANA Live is not relevant for S/4HANA.

SAP HANA Live Architecture

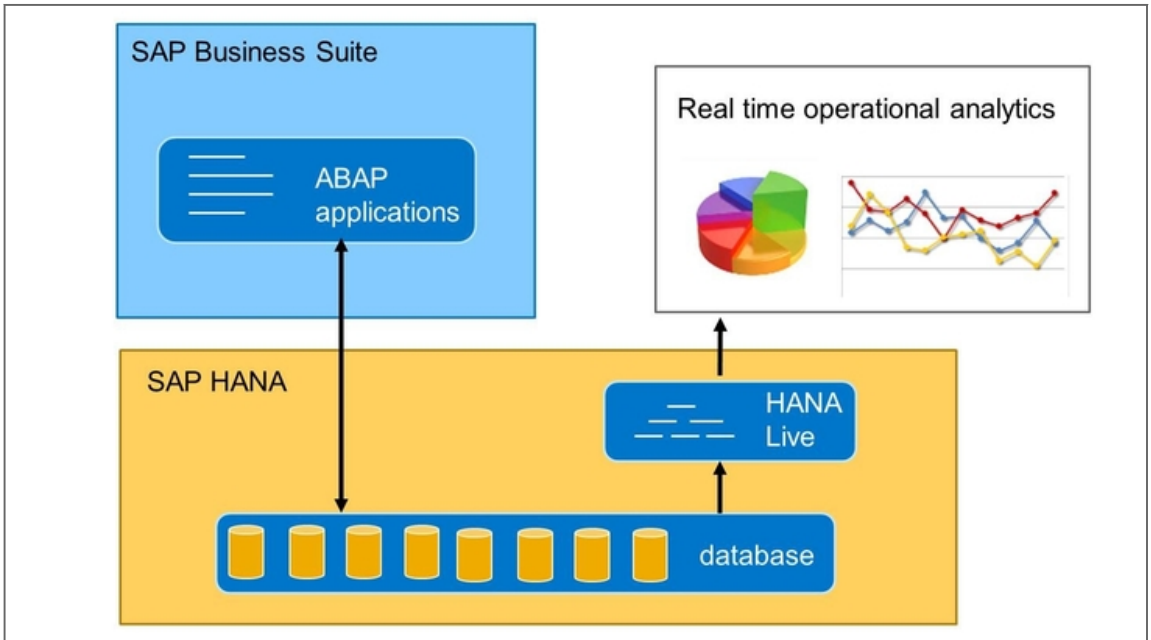


Figure 67: SAP HANA Live Architecture

Due to the high degree of normalization of the database schemas found in SAP Business Suite applications, tables can appear complex, fragmented, and are often difficult to consume

without a thorough understanding of each schema. SAP HANA Live provides ready-to-use business views of the Business Suite source tables. This means that report developers and application builders can easily consume live business information without being concerned about the underlying table complexities. The calculation views are created and maintained by SAP using standard SAP HANA modeling tools, so that they can be modified easily by customers using the same tools.

The key use case for SAP HANA Live is real-time operational analytics on SAP Business Suite. SAP HANA Live is a read-only virtual data model, so it is of no use for applications that need to write data back to the database. SAP HANA Live can be consumed immediately by SAP BusinessObjects reporting tools and, in fact, any clients that are able to consume standard SAP calculation views by SQL and MDX.

Key Aspects of SAP HANA Live

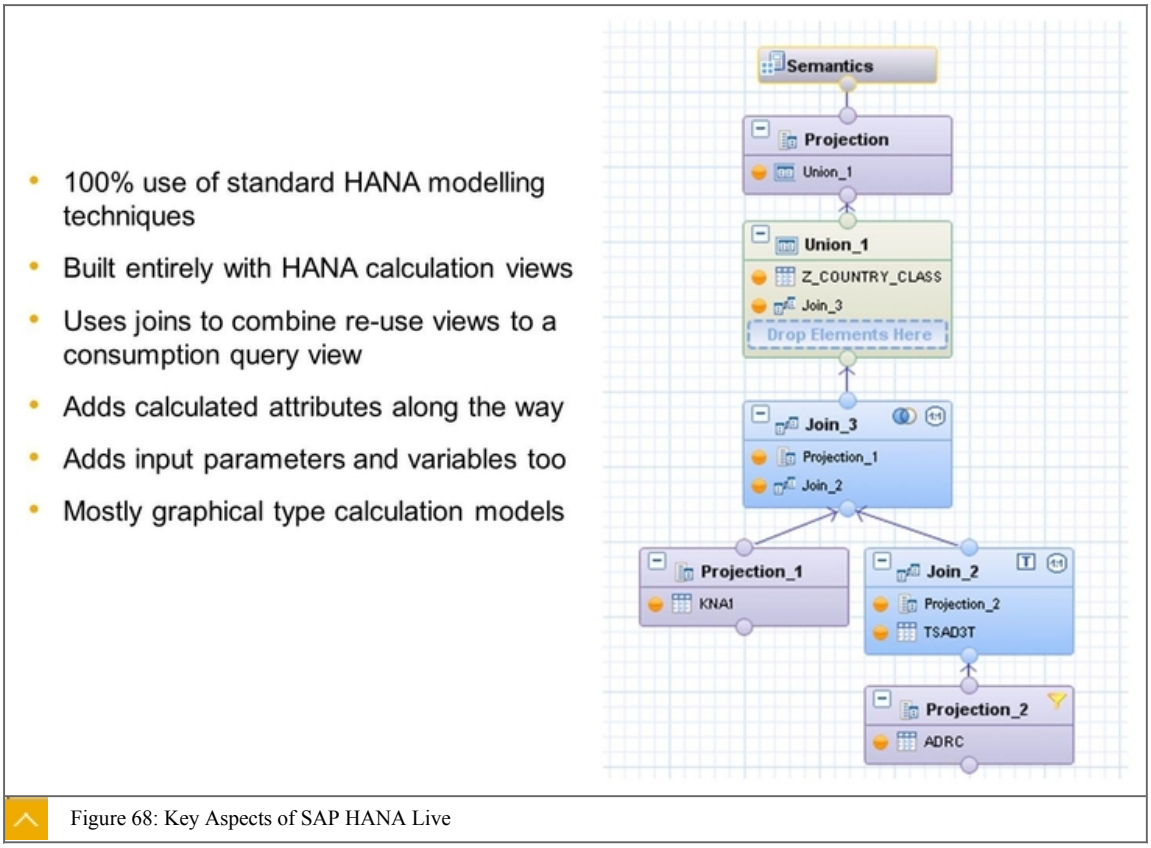


Figure 68: Key Aspects of SAP HANA Live

SAP HANA Live models contain all the necessary joins, filters, aggregations, and transformations to transform the data in the raw Suite tables into meaningful information. They are compiled into column views that are technically no different from the column views created from customer calculation views.

SAP HANA Live views do not aggregate data. They expose the data at the most granular level. However, the reporting tools are able to aggregate the data as required. They can be consumed by any SAP BusinessObjects reporting tool or any client that is able to consume a standard SAP HANA calculation view.

Most SAP HANA Live tools are integrated into SAP HANA Studio. However, some SAP HANA Live tools are built using SAPUI5 and are not part of Studio.

SAP HANA Live is optional and can be freely downloaded and installed into SAP HANA. It is not a licensed product.

SAP HANA Live has its own release cycle. So, check the versions available and evaluate which version will work with your SAP Business Suite and SAP HANA solution combination.

SAP HANA Live Deployment Scenarios

A key decision that customers must make is to decide whether to implement SAP HANA Live as an integrated scenario or as a side-by-side deployment.

Side-by-side deployment requires the implementation of a data loading tool, to move data from the tables in the source Suite application to SAP HANA. SAP Landscape Transformation Replication Server (SLT) is recommended, but other data provisioning tools can also be used. The key objective is to ensure that operational tables are replicated to SAP HANA. One of the most powerful tools used to explore SAP HANA Live models is the SAP HANA Live Browser.

The SAP HANA Live Browser is able to generate a list of tables required to support any SAP HANA Live model. One of the major benefits of using SLT for table replication is that it can consume the list of tables produced by SAP HANA Live Browser. This makes setting up replication easier.

Alternate Deployment Scenarios

The alternative is to enter all tables manually. Note that with a side car deployment, the data exposed by the SAP HANA Live view is only as up-to-date as the last replication of data from the source tables. So, if you set your replication to ten minute intervals, you do not really have a ‘live’ view of your source data.

Ideally, all operational data should be replicated in real-time to provide up-to-date information, hence the name SAP HANA Live.

With the integrated deployment scenario, SAP HANA Live is installed inside SAP HANA that is used to power the Business Suite application. This means that the tables used by the Business Suite applications are the same tables on which SAP HANA Live calculation views are based. No data movement takes place, so there is zero redundancy. This means that the SAP HANA Live views always reflect real-time information with no copying of data needed.

SAP HANA Live views supplied by SAP are technically no different to any other views in SAP HANA. This is true whether they are created by SAP, partners, or customers. As with any SAP HANA view, all SAP BI reporting tools are able to consume SAP HANA Live views. SAP HANA Live views frequently contain variables, and these are consumed by each SAP BI reporting tool. This means that you are able to provide popup filters to the users, to make personal data selections.

Core Data Services

Core Data Services (CDS) is a language used to define the data definition and the consumption model. As well as the technical definition of the data model and consumption model, CDS allows you to fully describe the business semantics right in the model. The semantics can be consumed by various applications and provide additional information about the data. For example, you can describe the correct way to aggregate a specific measure, or to identify certain fields as belonging to a customer address.

There are two types of CDS; HANA CDS and ABAP CDS.

HANA CDS

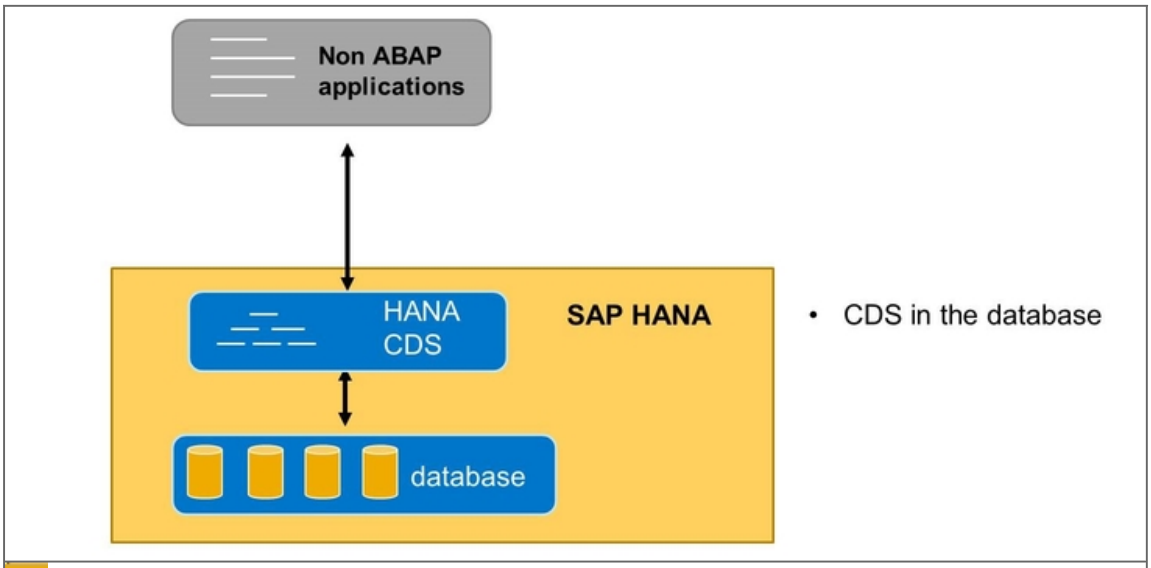


Figure 69: HANA CDS

HANA CDS is a component of SAP HANA. It sits on top of the SAP HANA database. HANA CDS is built using a programming language, very similar to SQL stored in CDS documents. The CDS documents are maintained in the SAP HANA Studio or Web IDE editors. They are physically stored in SAP HANA.

The overall aim of SAP HANA CDS is the same as SAP HANA Live — that is, to provide an easy-to-consume data model on top of complex table schemas.

SAP HANA Live has a single use case, which is to expose real-time operational data from SAP Business Suite tables in an easy-to-consume manner, to support analytics. In other words, SAP HANA Live's only use case is Business Intelligence (BI).

Difference Between HANA CDS and SAP HANA Live

HANA CDS is also able to achieve this, but HANA CDS is far more versatile and does more than SAP HANA Live. With CDS you not only expose tables for analytic purposes, but you can also use CDS in many other use cases, including search applications and planning applications. SAP HANA Live defines only the views on top of the existing database tables. In contrast, HANA CDS can actually be used to define the tables as well as the views.

Another key difference between HANA Live and HANA CDS is the method used to develop them. SAP HANA Live was created by SAP using the graphical SAP HANA modeling tools. HANA CDS can also be created using a graphical editor as well as a text-based scripting language, similar to SQL.

HANA CDS can be complex, not only to read, but also to construct, compared to the simpler graphical environment used by SAP HANA Live. This is especially the case when the HANA CDS is built using the text editor. Both approaches can sit side-by-side, but you will find few BI modelers using CDS and few application developers using SAP HANA Live. So, while it is true that HANA Live could have been created using HANA CDS, SAP chose to use the graphical modeling tools so that modelers would find it easy to understand, extend, and integrate the models.

ABAP CDS

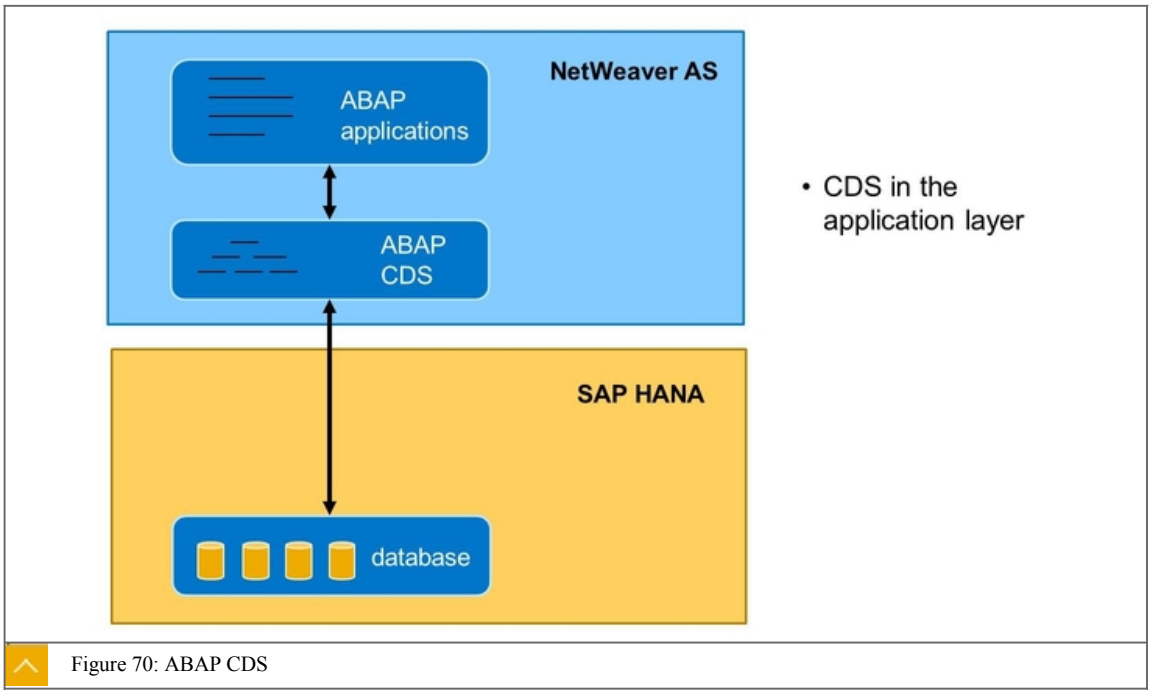


Figure 70: ABAP CDS

ABAP CDS is identical to HANA CDS at the conceptual level. However, the technical implementation is different, and the syntax of the language is different also.

ABAP CDS is part of the NetWeaver ABAP layer. It is used to define the data model used only by ABAP applications. You maintain ABAP CDS using ABAP tools alongside the ABAP application code in special CDS files.

As the name suggests, ABAP CDS is only relevant for ABAP applications and is part of the NetWeaver AS. Unlike HANA CDS, ABAP CDS can run on any database, not just SAP HANA. This means that once you create your ABAP CDS-based data model, you can redeploy it on any SAP supported database.

Importance of ABAP CDS

ABAP CDS allows you to define the data model and consumption model as a layer that sits between the database and the application logic. This provides great flexibility, which means no more locking in data modeling logic in the application, and no more locking the data modeling logic in the database.

SAP S/4HANA embedded analytics is based on ABAP CDS. SAP have built a complete virtual data model, using only ABAP CDS, that exposes the live operational tables from S/4HANA to all Fiori applications. Customers can also use the same ABAP CDS for their purposes.

HANA CDS and ABAP CDS are maintained using text editors. HANA CDS can also be maintained using a graphical editor in the Web IDE for SAP HANA.

CDS Editors

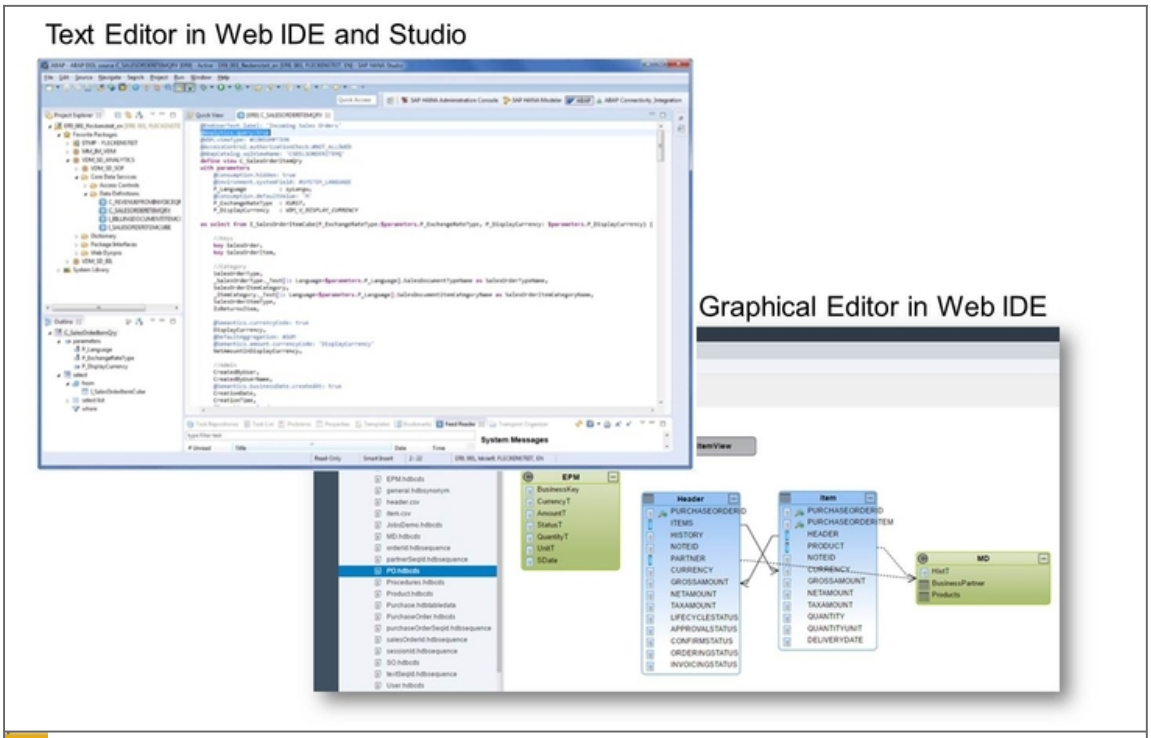


Figure 71: CDS Editors

LESSON SUMMARY

You should now be able to:

- Describe SAP HANA Live
- Describe Core Data Services (CDS)

Unit 3
Lesson 3

Defining SAP HANA Spatial Processing

LESSON OVERVIEW
In this lesson we will cover spatial processing with SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Define SAP HANA spatial processing

Developing Spatially-Enabled Business Applications

By combining business data with spatial data, you can identify new opportunities and root causes. You can do this by developing spatially-enabled business applications on the SAP HANA platform.

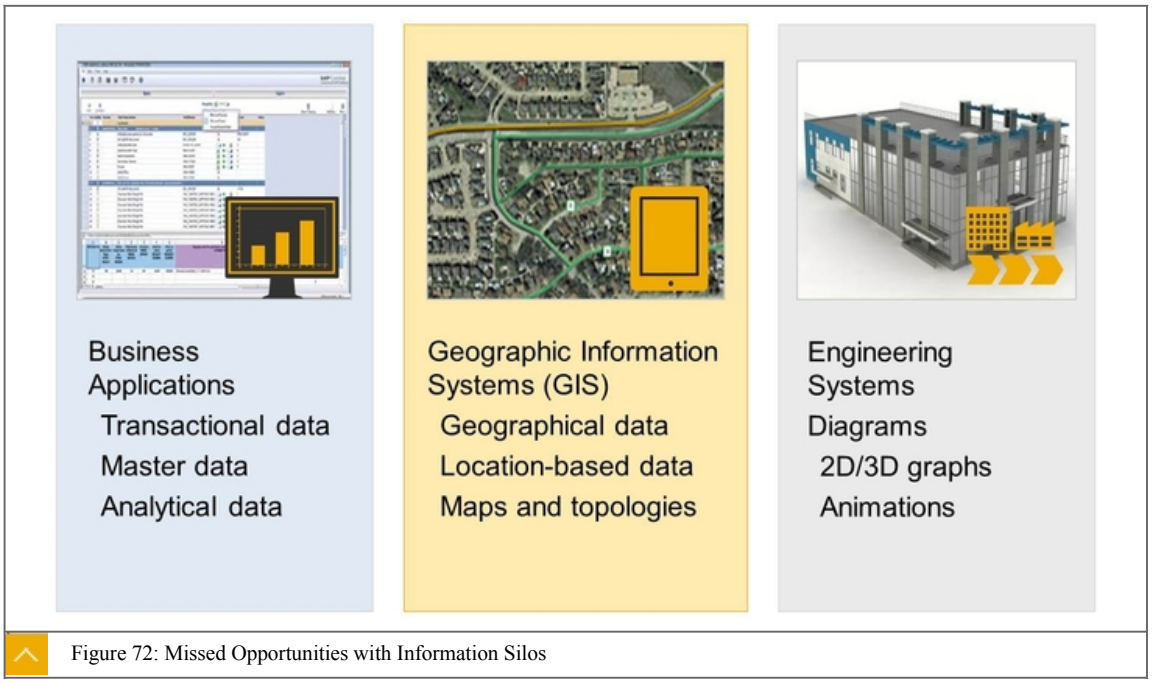


Figure 72: Missed Opportunities with Information Silos

Many organizations already rely on spatial data processing and use specialist applications alongside, but separate from, their business process applications. For example, a gas storage tank is leaking and an emergency repair order is raised in the SAP ERP system by a coordinator. Next, the coordinator uses a separate geo-based application and manually enters the asset number of the tank. The tank shows up on a map and the coordinator then uses another application to look up the nearest engineer, who is then dispatched.

Missed Opportunities with Information Silos

It would be more efficient if, at the time of generating the repair order, the ERP application was able to locate the tank. It could then identify and dispatch the nearest qualified engineer

who has enough working hours left to complete the repair, and provide useful geographic information to the engineer to describe how to best reach the tank. The application could then provide information about other equipment in the close vicinity that is due an inspection soon. This prevents having to make separate visits. This scenario could be possible if the core business processes were integrated with spatial data and analysis.

Beyond business applications, there are exciting use cases for spatial analysis in the sports environment. SAP have developed a series of applications that provide deep analysis of player performance. For example, in golf, by adding a sensor to the ball and pin, we can create a graphical history to illustrate the improvements in accuracy of the shot. These types of applications are already in use by major sports organizations around the world.

There are many applications that could be dramatically enhanced with the integration of spatial data.

Note:
Be careful to use the correct term spatial and not geographic or geo. Spatial processing in SAP HANA is not limited to just geographical scenarios, but can handle anything regarding the three dimensions of space.

SAP HANA Spatial Processing

SAP HANA includes a multilayered spatial engine and supports spatial columns, spatial access methods, and spatial reference systems. Spatial data is data that describes the position, shape, and orientation of objects in a defined space. Spatial data is represented as geometries in the form of points, line strings, and polygons. Take for example, a map of a state, representing the union of polygons representing zip code regions.

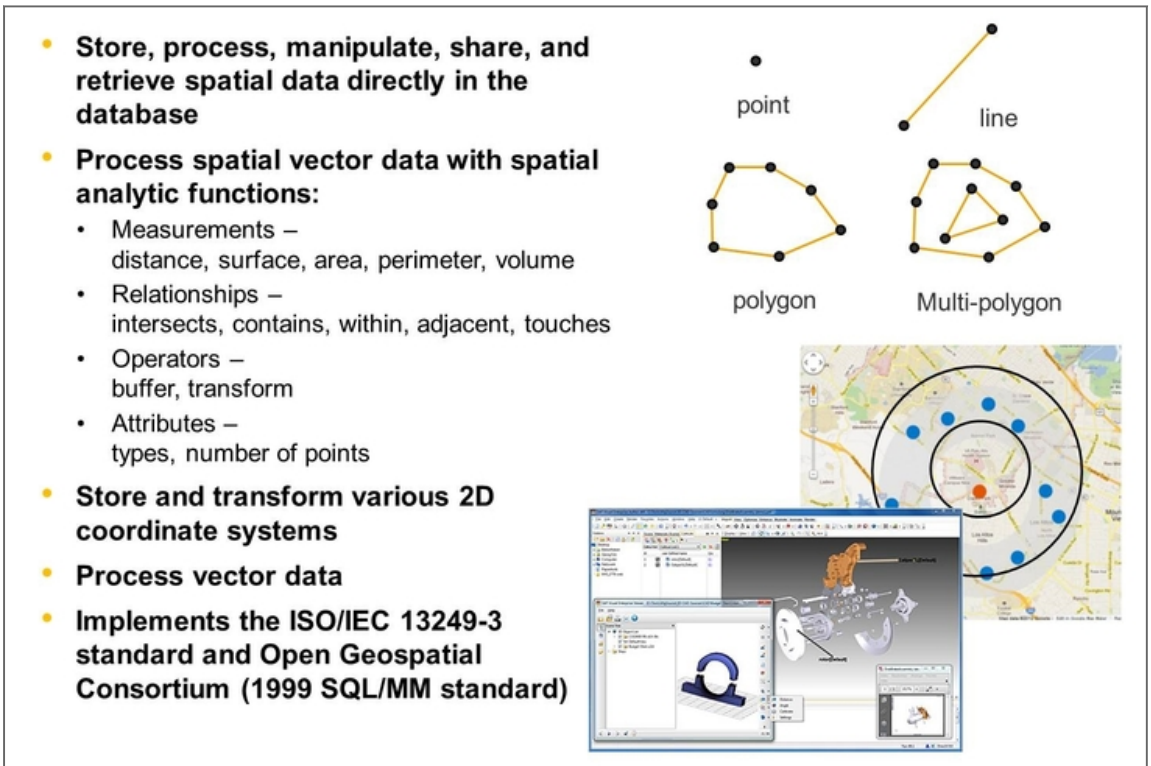


Figure 73: SAP HANA: Spatial-Enabled Database

SAP HANA provides storage and data management features for spatial data. This allows you to store information such as geographic locations, routing information, and shape data. These information pieces are stored as points and various forms of polygons and lines in columns defined with a corresponding spatial data type (such as ST_Point or ST_Polygon). You can then use methods and constructors to access and manipulate the spatial data.

For SAP HANA 2.0, we have developed a series of microservices that provide powerful spatial data for earth observation. These were sourced from the European Space Agency (ESA). Customers can now combine satellite images, and extract metadata from the images, and use them in their spatial applications. Already, customers are developing new applications to analyze and combine data relating to environmental conditions.

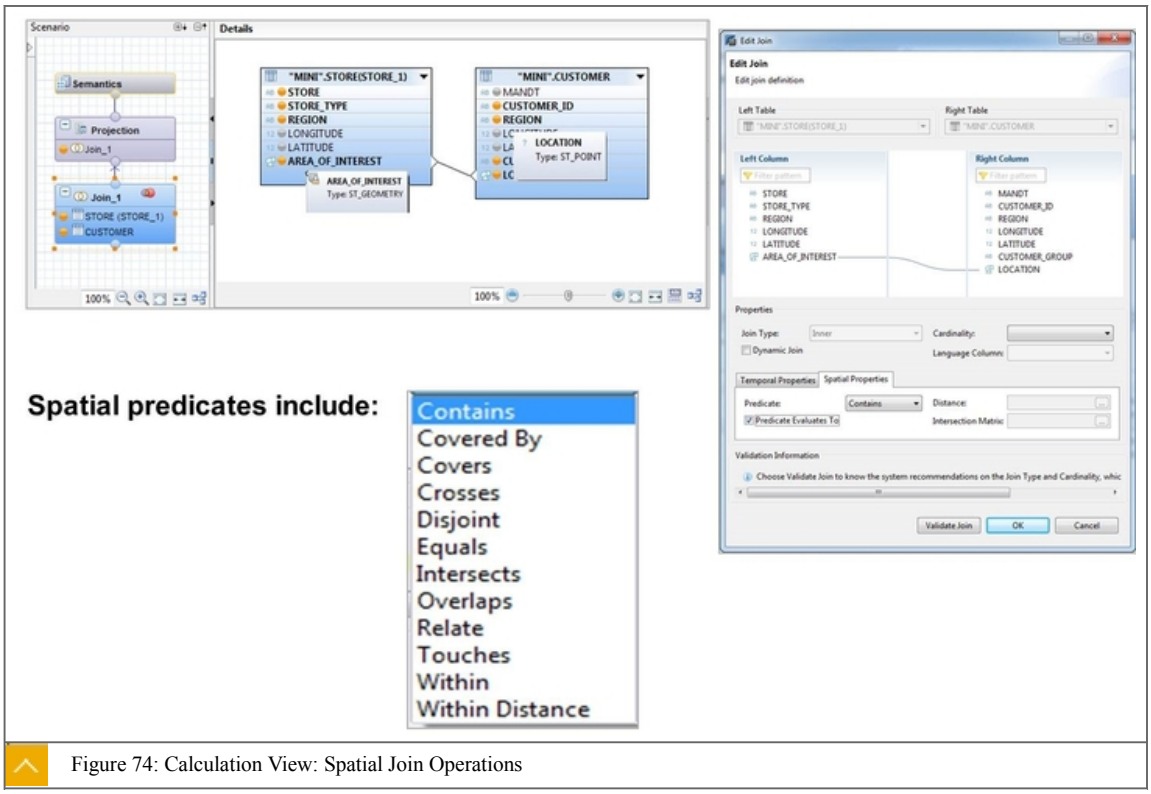


Figure 74: Calculation View: Spatial Join Operations

Calculation views now support spatial operations. These operations could include calculating the distance between geometries and determining the union or intersection of multiple objects. These calculations are performed using predicates such as intersects, contains, and crosses.

Note:
Spatial processing is covered in more detail in the course, HA300.

LESSON SUMMARY
You should now be able to:

- Define SAP HANA spatial processing

Unit 3
Lesson 4

Defining Text Analytics

LESSON OVERVIEW
In this lesson we will cover text processing with SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Define text analytics

Text Analytics with SAP HANA

A typical challenge in today's business world is to search for information among huge amounts of unstructured data. This can occur in different areas, inside or outside your company. You may then need to extract meaningful information from your unstructured data, to be able to analyze it properly.

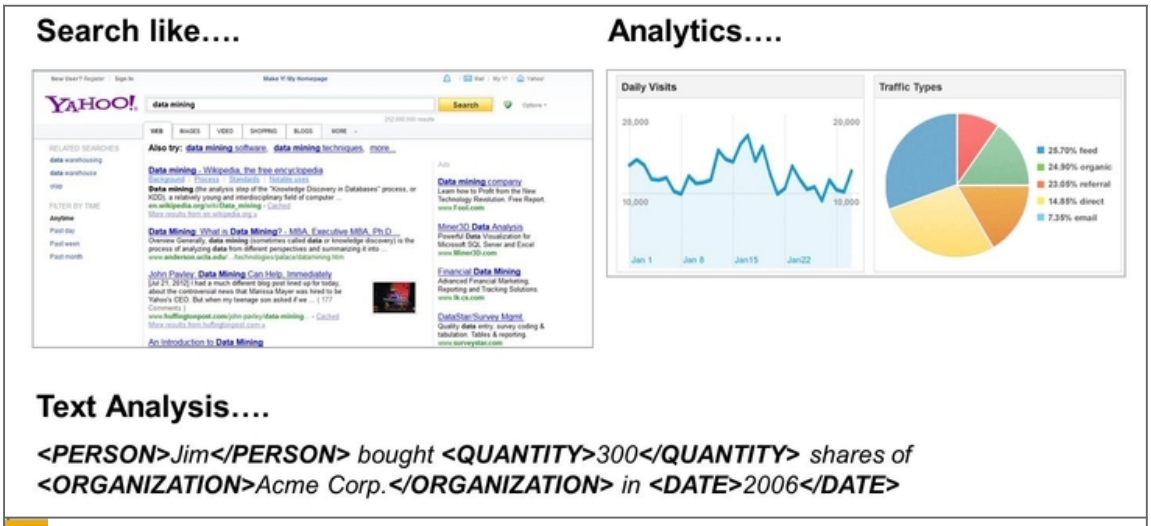


Figure 75: Text Search, Text Analysis, and Analytics

SAP HANA has powerful in-memory text search capabilities. You can search through huge amounts of unstructured data quickly and identify key entities such as company names, product names, and sentiments of customers. You may also have additional search requirements, like ranking your search results or bringing back a snippet of text that surrounds the found word, to put it into context. SAP HANA offers native search capabilities to support these and other business needs with the full Text Search, Text Analysis, and Text Mining.

SAP HANA Text Search

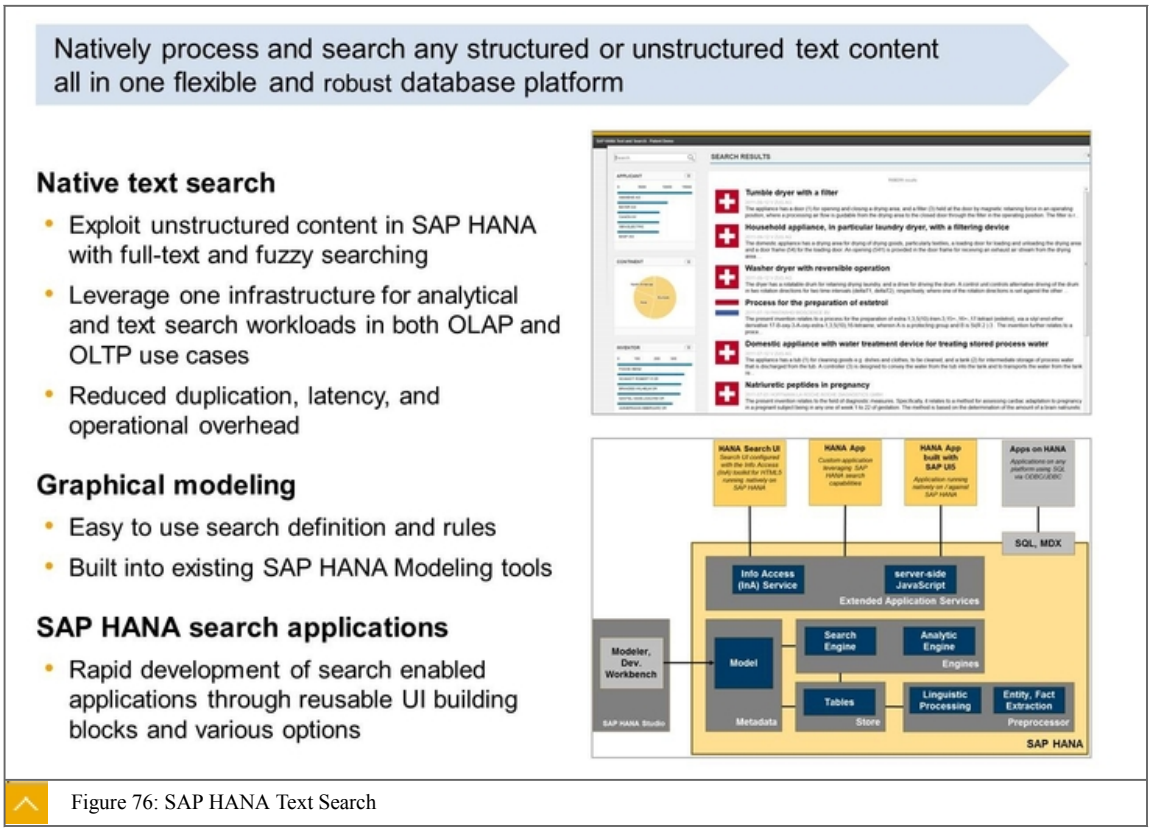


Figure 76: SAP HANA Text Search

Among the search capabilities in SAP HANA is the Fuzzy Search. This function offers the possibility of searching for approximate matches to a given criterion. For example you could search for the string **`Walldorf`** and find all exact and approximate matches (to a degree that you can specify in your search) . These matches could include results like **`Walldorf`**, **`Wadlorf`**, **`Valldorf`**, and **`Wahldorff`**.

In general, a Fuzzy Search will search for all criterion matching the given one, either exactly, or differing by missing, added, or mistyped characters. The behavior is similar to that of a web search engine; that is, you do not want to see only exact matches of your search criterion.

SAP HANA Text Analysis

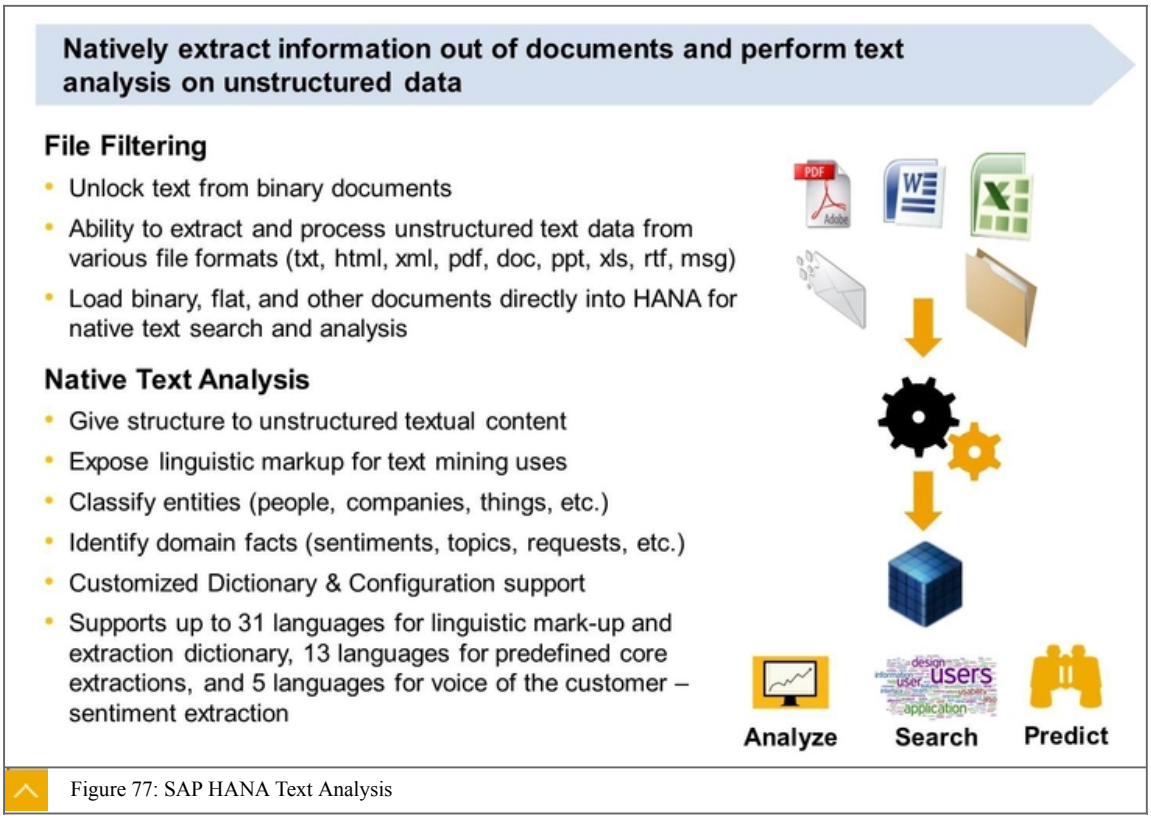


Figure 77: SAP HANA Text Analysis

Text Analysis allows the extraction of structured information from unstructured information. An example of this is linguistic markup, which entails identifying the various parts of a speech (verbs, nouns, adjectives, and so on). It also allows you to identify entities (locations, persons, and dates) in an unstructured text.

The results of a Text Analysis are stored in a table and can therefore be consumed through all supported SAP HANA scenarios. This includes information models, R language scripts, the Predictive Analytic Library functions, or the Suite on HANA Information Access toolkit for HTML5 by building a search UI.

SAP provides ready-to-go reference dictionaries of entities (company names, dates, countries, and so on) as well as a dictionary called “Voice of Customer”. This stores known words and expressions that customers use to express their feelings about products and services. It can be used in sentiment analysis to help organizations quickly respond to customer feedback on social media and other places where textual feedback is found. Customers can also create their own custom dictionaries to include words and expressions used in their organizations.

Text Search and Analysis Use Cases/Scenarios

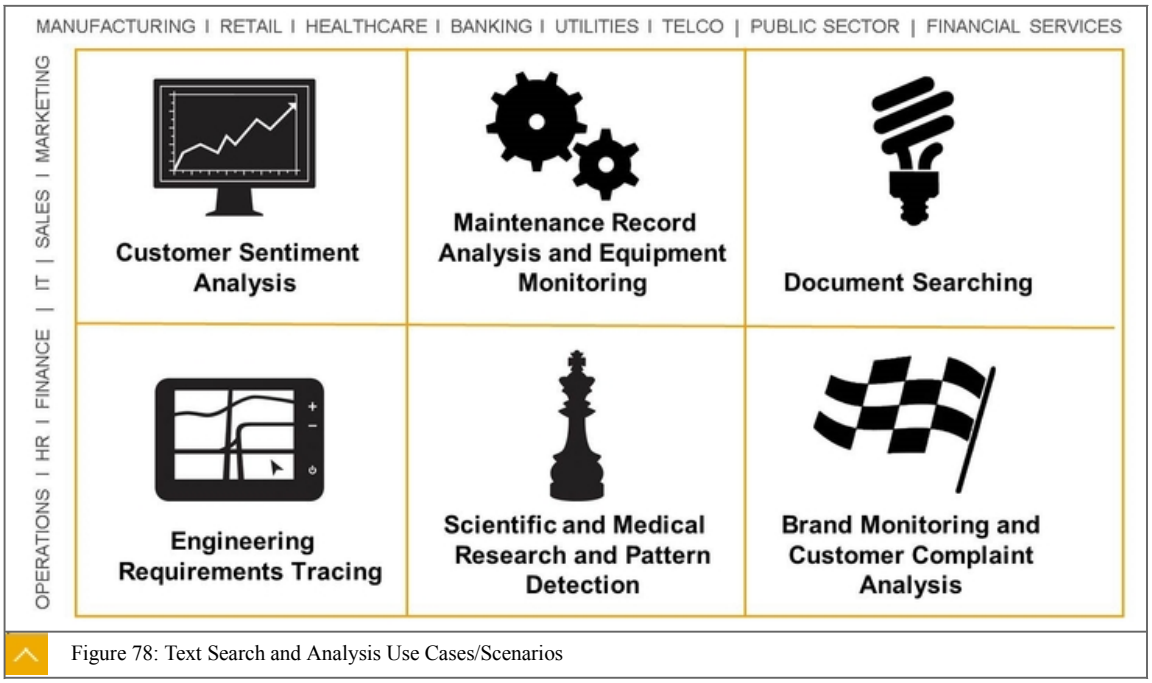


Figure 78: Text Search and Analysis Use Cases/Scenarios

LESSON SUMMARY

You should now be able to:

- Define text analytics

Unit 3
Lesson 5

Explaining Predictive Modeling

LESSON OVERVIEW
In this lesson we will cover predictive modeling with SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe predictive analysis in SAP HANA

Predictive Analysis

Predictive Analysis is a new breed of advanced analytical applications that provide insight by processing large amounts of data. They process both historical and current data, using powerful algorithms to generate probable outcomes. Traditionally, predictive analysis was not possible for most organizations due the following reasons:

- Lack of quality and quantity of data
- Not enough powerful processing resources
- Algorithms that were too generic and produced weak results
- Lack of skills in developing predictive models and understanding the results

With the phenomenal growth of interest in Data Science, organizations are realizing that they need to quickly develop skills in this area. SAP HANA plays a key role in Data Science, providing the platform for in-memory data management and processing, especially in the provision of sophisticated algorithms. SAP HANA provides a large library of advanced, built-in algorithms and powerful in-memory processing capabilities. It also provides the ability to easily reach and consume Big Data.

Internet of Things (IoT) scenarios often require embedded analytical processes to provide automated, real-time decision making and machine learning.

There are many use cases for predictive analysis.

Predictive Analysis Examples

Figure 79: Predictive Analysis Examples

Predictive Analysis Algorithms

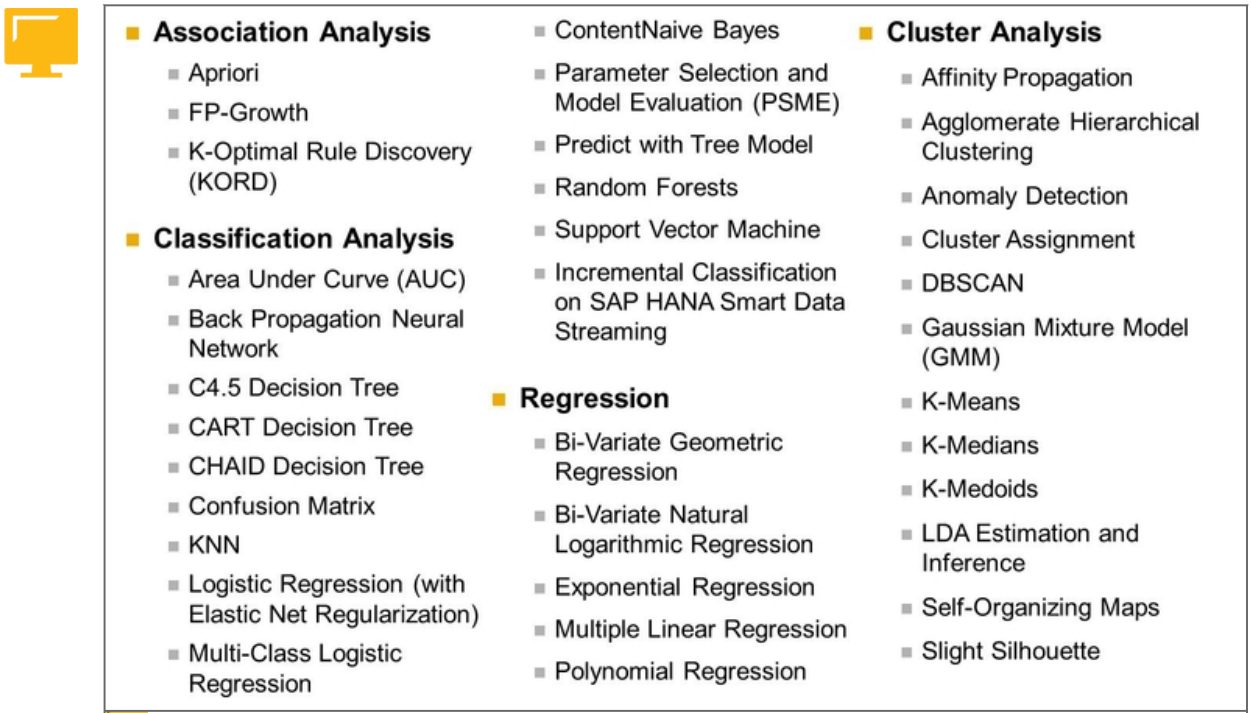


Figure 80: Predictive Analysis Algorithms

SAP HANA has a large library of built-in algorithms that are ready for use. It is also possible to import more algorithms from the huge, public R library and to develop your own custom algorithms in the R language.

Note:
R is a statistical scripting language, which is heavily used in academic and scientific institutions. It is becoming more widely used, especially in the commercial sector, and data scientists are often fluent in this language.

The list of built-in algorithms is growing all the time. There are algorithms for all types of analysis and they can be grouped into families. Some of these families are as follows:

- Associations

  When this happens, what else happens?

- Classification

  People like this, usually do this

- Cluster

  These customers exhibit similar buying behavior

- Regression

  Customers with income x and children y, usually spend z

- Time series

  In the last few years sales are falling but at this time of year we still expect to sell x amount

- Outlier detection

  These sports fans are not behaving like the rest of their group

- Weighted scoring

  Vendors who deliver late get a big penalty and a lesser penalty for wrong pricing

Create a Predictive Model

To get started with predictive analysis on SAP HANA, you must install the Predictive Analysis Library (PAL). PAL contains the catalog of algorithms and data preparation tools developed by SAP.

Then, using either Studio or Web IDE for SAP HANA, you create a flowgraph. This is a graphical definition of the data sources, data preparation steps, the algorithms, and the output.

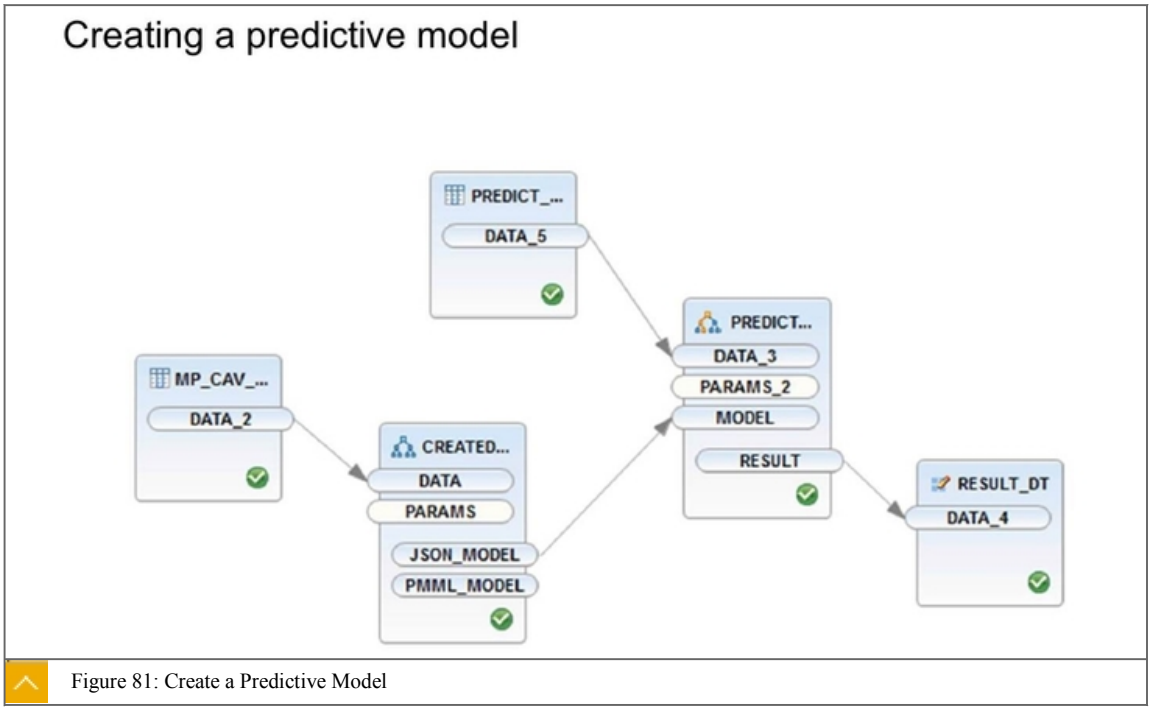


Figure 81: Create a Predictive Model

The flowgraph generates an object that can be scheduled, triggered by an event, or embedded in a calculation view.

LESSON SUMMARY
You should now be able to:

- Describe predictive analysis in SAP HANA

Unit 3
Lesson 6

Describing Graph Processing

LESSON OVERVIEW
This lesson introduces graph processing with SAP HANA and the key role it plays in data modeling.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe graph processing with SAP HANA

Graph Modeling

Graphs are used to model data when it is best represented using a network. Examples of use cases include logistics and transportation, utility networks, and social networks. The basic idea behind graph modeling is that it allows a modeler to define a series of entities (nodes) and link them in a network. This network represents how they relate to each other.

Graph models can indicate flow direction between entities. This means that additional meaning can be added to the network and traces can be made. Imagine a complex supply chain mapped using a graph, where all manufacturers, suppliers, distributors, customers, and consumers are represented with information stored along the connections. The benefit to this form of modeling is that it makes it easy to develop applications that can traverse huge graphs at speed. As a result you can ask questions such as the following:

- How many hours has the product traveled between two specified points in the network?
- Where are all the possible points of origin of this product?
- Describe the entire journey of a product by listing all of the stop off points.

Graph processing allows you to discover hidden patterns and relationships in your data, and all in real time.

Graph Model: Example

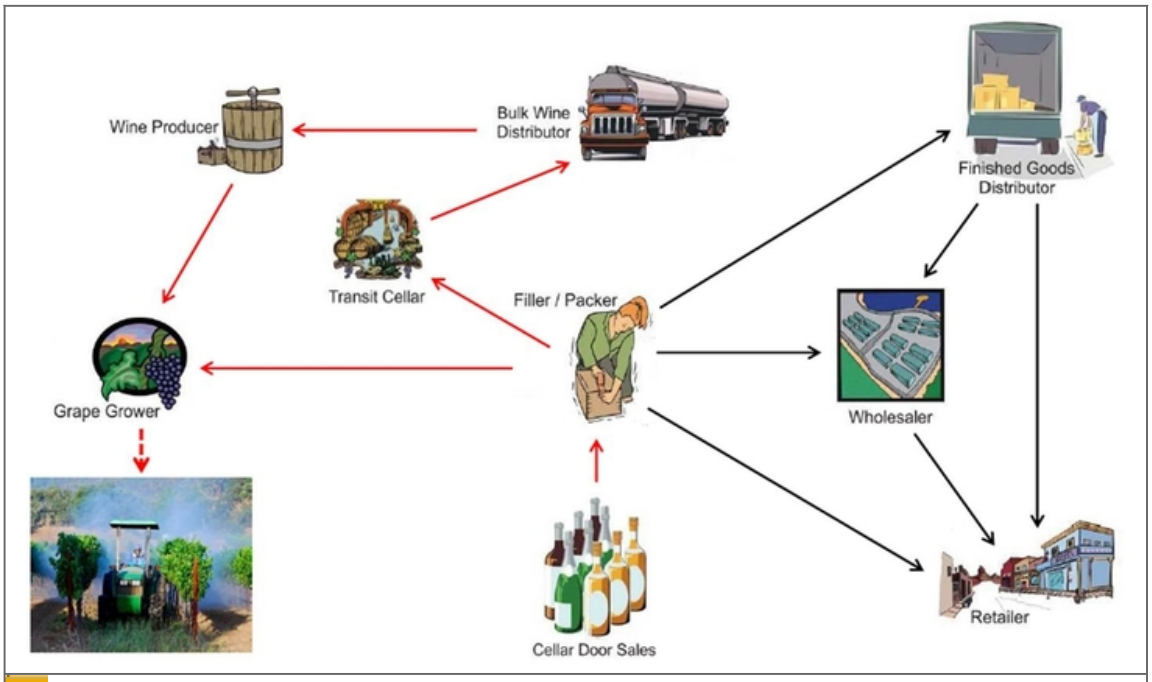


Figure 82: Graph Model: Example

The graph model in the figure, Graph Model: Example , is one that most people can relate to. However, there are many other interesting examples, including the following:

- Medical

  Create a network of patients, conditions, treatments, and outcomes for reuse in diagnosis and planning treatments of other patients.

- Social Network

  Using popular social media portals, find your customers and their friends, friends of friends, and likes or dislikes to create marketing opportunities.

It is possible to use standard SQL data definitions and query syntax to create and process a similar model. However, it would be extremely complex in the definition of the model and the SQL needed for the querying of the graph, plus processing times could be challenging. SAP HANA Graph provides tools for graph definition and language for graph processing, to ensure that model development is more natural and simplified. It also guarantees that the processing is flexible, and of course, optimized for in-memory processing.

Graph Modeling with SAP HANA

The key objects in a graph model are the vertices and edges. Vertices are stored in tables and represent the members of the graph. These are the nodes. The edges are also stored in tables and describe the lines between the members. Along each line, you can store connection information such as distance, preference, strength, and so on.

To get started with SAP HANA Graph, you first need to create and fill the vertices and edge tables with data. You then create a Graph Workspace that refers to the vertices and edge tables. The Graph Workspace simply creates the metadata that defines the graph model and how the vertices and edge tables relate. No actual business data is stored in a Graph Workspace.

The Graph Viewer is a native SAP HANA application that provides an interface to interact with and visualize graph workspaces in SAP HANA. SAP HANA Graph Viewer is an additional tool for SAP HANA Graph that can be downloaded from the SAP Software Download Center.

SAP HANA Graph Viewer

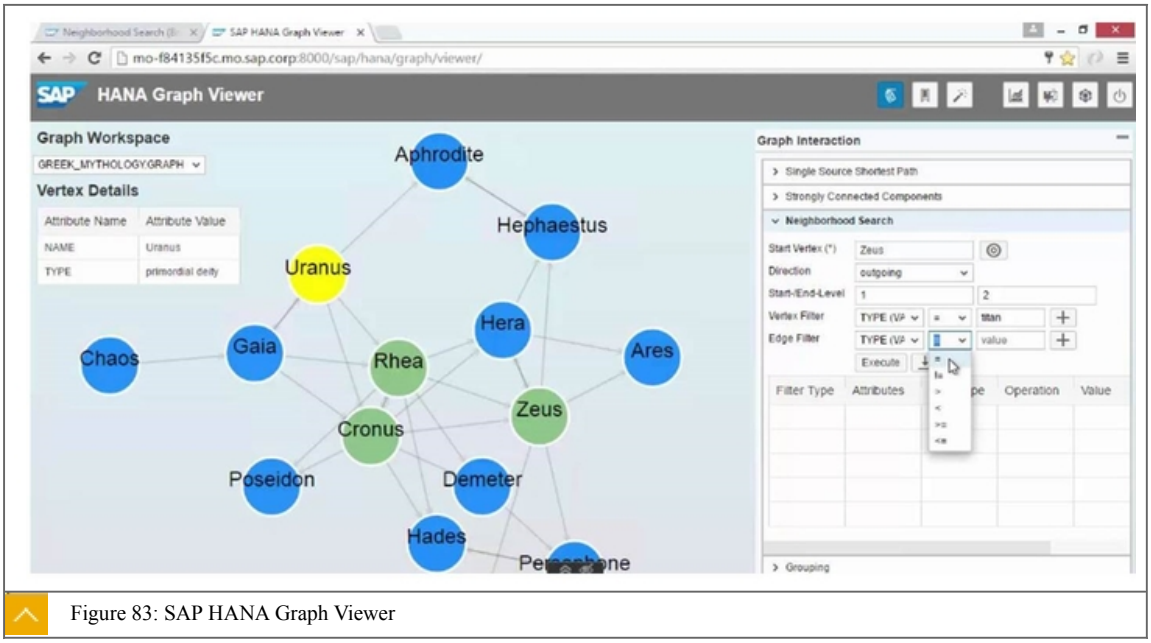


Figure 83: SAP HANA Graph Viewer

SAP HANA Graph provides a native processing language for graph query and manipulation called WIPE (weakly structured information processing and exploration). It is a declarative language, similar to SQL. It contains special keywords to allow easy graph query questions to be formulated such as, “how far?”, “how deep?”, “what’s the source?”, “where is the end point?”, “where is the strongest connection?”, and “what is the shortest path?”.

SAP HANA Graph utilizes a dedicated graph engine that works with the other data processing engines in SAP HANA. SAP HANA Graph processing can be combined with all other types of SAP HANA data processing so that sophisticated and innovative applications can be developed on any data types.

LESSON SUMMARY
You should now be able to:

- Describe graph processing with SAP HANA

Unit 3

Learning Assessment

1. What is the role of core modeling in SAP HANA?

   Choose the correct answers.

   - ☐ **A** To push data-intensive processing up to the application to obtain the best performance
   - ☐ **B** To develop reuseable data processing logic in the database
   - ☐ **C** To simplify applications

2. What are valid types of calculation view?

   Choose the correct answers.

   - ☐ **A** Dimension
   - ☐ **B** Cube
   - ☐ **C** Cube with star schema
   - ☐ **D** Dimension with star schema

3. If I need to create multiple, tabular output structures from an SQLscript, which HANA modeling object do I use?

   Choose the correct answer.

   - ☐ **A** Procedure
   - ☐ **B** Function

4. What do we implement to restrict access to specific data rows of a calculation view?

   Choose the correct answer.

   - ☐ **A** SQL permission
   - ☐ **B** Authorisation object
   - ☐ **C** Analytic privilege

5. Which SAP solution is used with SAP HANA Live?

   Choose the correct answer.

   - ☐ **A** SAP S/4HANA
   - ☐ **B** SAP Business Suite
   - ☐ **C** SAP Ariba

6. Which of these is used to build the virtual data model for SAP S/4HANA?

   Choose the correct answer.

   - ☐ **A** ABAP CDS
   - ☐ **B** SAP HANA Live
   - ☐ **C** Universe
   - ☐ **D** SAP HANA CDS

7. What SAP HANA spatial type would you most likely use to represent the State of Colorado on a visual application?

   Choose the correct answer.

   - ☐ **A** Polygon
   - ☐ **B** Line
   - ☐ **C** Point

8. In SAP HANA Text Analysis, why would you use the 'Voice of Customer' dictionary?

   Choose the correct answer.

   - ☐ **A** To extract the sentiment from customer feedback on social media
   - ☐ **B** To extract common entities such as company, country, currencies, and so on, found in documents
   - ☐ **C** To identify close matches in words and expressions, to catch misspellings

9. What must I create to develop a predictive model?

   Choose the correct answer.

   - ☐ **A** Calculation View
   - ☐ **B** Predictive Analysis Library
   - ☐ **C** Flowgraph
   - ☐ **D** Procedure

10. SAP HANA Graph Processing powers real-time, high-performance business charts and dashboards.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

Unit 3

Learning Assessment - Answers

1. What is the role of core modeling in SAP HANA?

   Choose the correct answers.

   [ ] **A** To push data-intensive processing up to the application to obtain the best performance

   [X] **B** To develop reuseable data processing logic in the database

   [X] **C** To simplify applications

   You are correct! We push data intensive processing down to the database to obtain the best performance. Yes, core modeling in HANA encourages reuse of processing logic. Yes, we simplify applications when we don't have to worry about developing code to carry out data processing tasks. Refer to HA100, Unit 3, Analytical Processing with SAP HANA , for details.

2. What are valid types of calculation view?

   Choose the correct answers.

   [X] **A** Dimension

   [X] **B** Cube

   [X] **C** Cube with star schema

   [ ] **D** Dimension with star schema

   You are correct! Dimension with star schema is the only type that is not valid. Refer to HA100, Unit 3, Analytical Processing with SAP HANA , for details.

3. If I need to create multiple, tabular output structures from an SQLscript, which HANA modeling object do I use?

   Choose the correct answer.

   [X] **A** Procedure

   [ ] **B** Function

   You are correct! Functions can only produce one tabular output whereas Procedures can produce multiple tabular outputs. Refer to HA100, Unit 3, Analytical Processing with SAP HANA, for details.

4. What do we implement to restrict access to specific data rows of a calculation view?

   Choose the correct answer.

   ☐ **A** SQL permission

   ☐ **B** Authorisation object

   ☒ **C** Analytic privilege

   You are correct! SQL permissions only secure the database object, not data values. Authorisation object is an ABAP object, not used in SAP HANA. Analytic Privilege is used to define security around rows (data). Refer to HA100, Unit 3, Analytical Processing with SAP HANA, for details.

5. Which SAP solution is used with SAP HANA Live?

   Choose the correct answer.

   ☐ **A** SAP S/4HANA

   ☒ **B** SAP Business Suite

   ☐ **C** SAP Ariba

   You are correct! SAP HANA Live is used only with SAP Business Suite. Refer to HA100, Unit 3, Virtual Data Models with SAP HANA , for more details.

6. Which of these is used to build the virtual data model for SAP S/4HANA?

   Choose the correct answer.

   ☒ **A** ABAP CDS

   ☐ **B** SAP HANA Live

   ☐ **C** Universe

   ☐ **D** SAP HANA CDS

   You are correct! S/4HANA uses ABAP CDS for its virtual data model. Refer to HA100, Unit 3, Virtual Data Models with SAP HANA , for more details.

7. What SAP HANA spatial type would you most likely use to represent the State of Colorado on a visual application?

   Choose the correct answer.

   [X] **A** Polygon

   [ ] **B** Line

   [ ] **C** Point

   You are correct! A State, being an area, would best represented with a polygon spatial type. These can be used to describe any type of shape. Points are great for identifying entities that do not occupy visual space, such as customer or hotel. Lines are great for representing something continuous that has a start and end such as a road or river. Refer to HA100, Unit 3 Analytical Processing for details

8. In SAP HANA Text Analysis, why would you use the 'Voice of Customer' dictionary?

   Choose the correct answer.

   [X] **A** To extract the sentiment from customer feedback on social media

   [ ] **B** To extract common entities such as company, country, currencies, and so on, found in documents

   [ ] **C** To identify close matches in words and expressions, to catch misspellings

   You are correct! 'Voice of Customer' is a standard dictionary used in sentiment analysis that contains expressions that customers may use to describe how they feel about products or services: Refer to HA100, Unit 3, Analytical Processing with SAP HANA , for more details.

9. What must I create to develop a predictive model?

   Choose the correct answer.

   [ ] **A** Calculation View

   [ ] **B** Predictive Analysis Library

   [X] **C** Flowgraph

   [ ] **D** Procedure

   You are correct! A predictive analysis model is created with a Flowgraph either in the Studio or Web IDE. Refer to HA100, Unit 3 Lesson 5 for more details.

10. SAP HANA Graph Processing powers real-time, high-performance business charts and dashboards.

    Determine whether this statement is true or false.

    [ ] True

    [X] False

    You are correct! Graph Processing is not related to charts and dashboards, but is used to model and process data that is best described using a network. Examples would be supply chains or social networks where many entities are highly connected. Refer to HA100, Unit 3, Analytical Processing with SAP HANA , for details.

UNIT 4 Data Management with SAP HANA

UNIT OBJECTIVES

- Describe SAP HANA data management
- Describe SAP HANA data acquisition solutions

Unit 4
Lesson 1

Describing Data Management

LESSON OVERVIEW
In this lesson we will explain what is meant by data provisioning with SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe SAP HANA data management

Data Management in SAP HANA

Data management in SAP HANA covers all topics related to data acquisition and data storage. One of the key strengths of SAP HANA is its powerful data connection capabilities. With SAP HANA you can build innovative applications that can access any data of any type, anywhere.

SAP HANA is able to extract and store data but also create connections to remote data sources for live data access. This means that you can decide whether you need to actually physically capture and store a copy of the data in HANA, or if you just need to access remote data.

You need to decide which data acquisition tools are required. There are many to choose from. You also need to decide where the data will reside. There are multiple options for data location in HANA. You must pay special attention to the architecture of your data management solution in SAP HANA and design this well for optimum performance at the lowest cost, especially for high volume data scenarios.

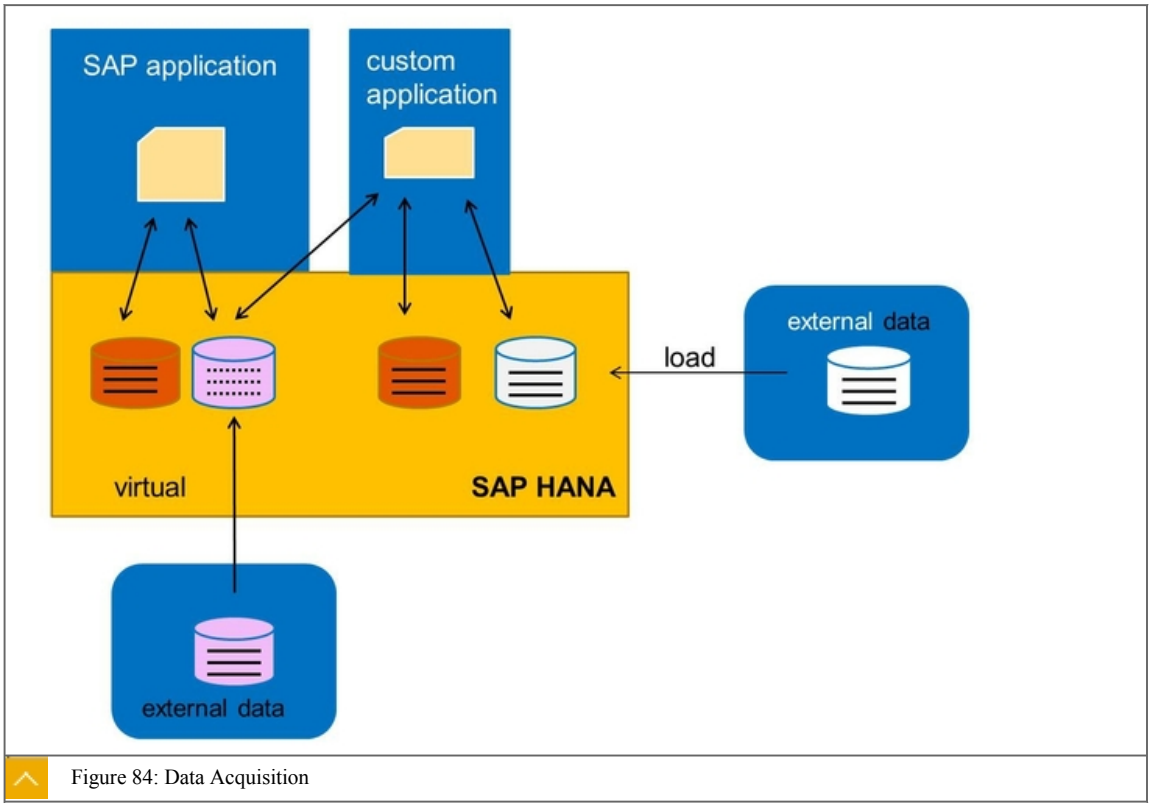


Figure 84: Data Acquisition

Applications that run on SAP HANA generate SQL statements that create, update, and delete records in the HANA database. In this case, no additional data acquisition tools are needed. This is true for SAP applications such as S/4HANA and BW and also for custom applications that run on HANA.

However, it is important to understand what happens when SAP HANA is deployed as a standalone database, for example, as a data mart with analytics running on top. Even if SAP HANA is powering applications such as S/4HANA or BW, you may want these applications to also have access to additional data sources, for example adding textual data from Twitter to a sales opportunity in ERP. This is why you implement the data acquisition tools of SAP HANA.

Data acquisition is not just another term for data loading. Data loading implies that data is physically moved and stored in SAP HANA. While this is possible, data acquisition also includes other approaches, such as data streaming and data virtualization. With these approaches, data is exposed to SAP HANA for processing, but data is not physically stored in SAP HANA.

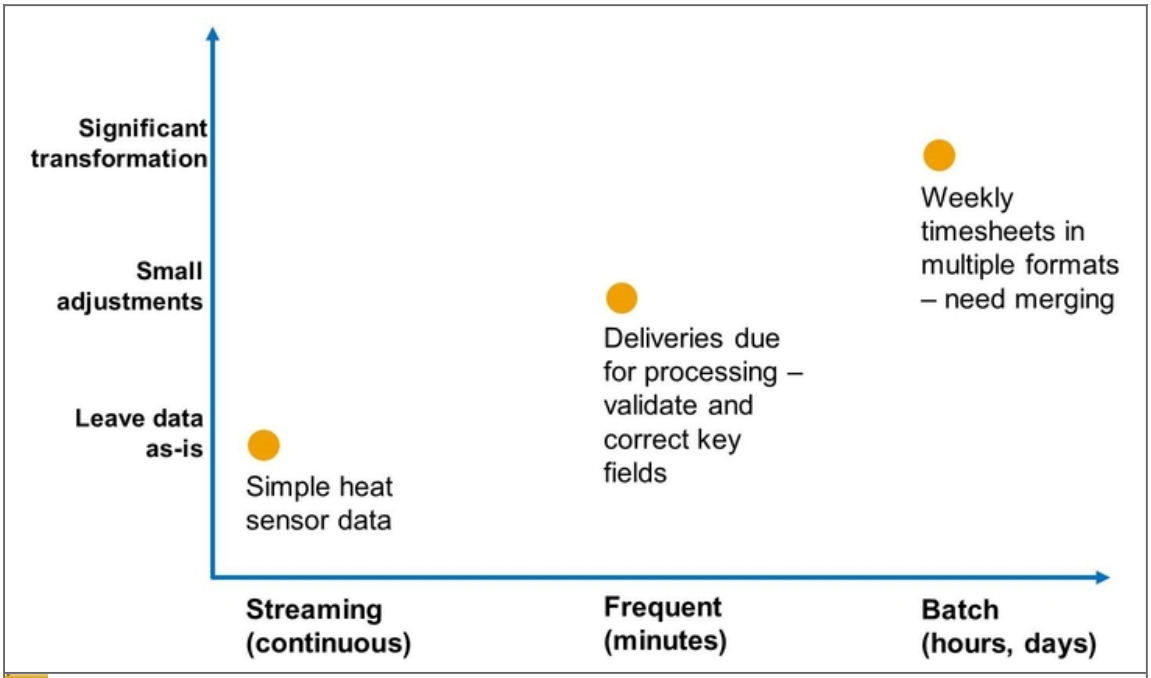


Figure 85: Types of Data Acquisition

When you consider data acquisition, think about the following key aspects:

- Frequency of data acquisition
- Transformation of data

Frequency is represented on the horizontal axis of the figure, Types of Data Acquisition . Data can arrive at different time frequencies, ranging from real-time and hourly, to weekly, or yearly. It could also be driven by occasional events, like when a vending machine runs out of stock and transmits a request for a refill.

Transformation is represented on the vertical axis of the figure, Types of Data Acquisition . During provisioning, data can be transformed. This transformation could be done to align billing codes from different systems, convert currencies, look up missing zip codes, or calculate rebate values. Sometimes data from different sources must be loaded at the same time, for technical or business integration logic to be applied. An example of this is when data needs to be harmonized into a single stream from different sales order entry systems.

Data Provisioning Tools

SAP HANA allows any combination of provisioning frequency with any degree of transformation. This enables you to meet the needs of all applications that require data at any speed and of any type.

Today's modern applications are powered by a rich variety of data types (transactional, spatial, and text). These applications consume data at different rates from continuous real-time sensor data, to periodic batch loads of bulk data.

The course, HA350, HANA Data Provisioning covers various data provisioning tools in more detail. HA355, SDI and SDQ provides a deep dive into SDI and SDQ.

Data Storage

As SAP HANA evolves, there are many new options for data storage across tiers. You can decide, based on the importance of the data and performance expectations, exactly where data should reside, in or even outside HANA. There are many choices available for your data

storage. You can choose between memory, disk, or external data archives. You can decide to implement Extended Storage or perhaps to implement multistore tables that cross tiers. You can classify your data by temperatures and assign the various temperatures to physical storage options in HANA.

Even though there is now vast amounts of memory available, it does not make sense to store old, rarely accessed data in memory. Data has a lifecycle and older data should be moved away from memory (hot), down to disk (warm), and eventually to archive (cold). The main reason for this is to ensure that you have the best possible performance for the data that is most used. You do not want to clog memory with huge amounts of old data that never gets used.

Data Temperatures

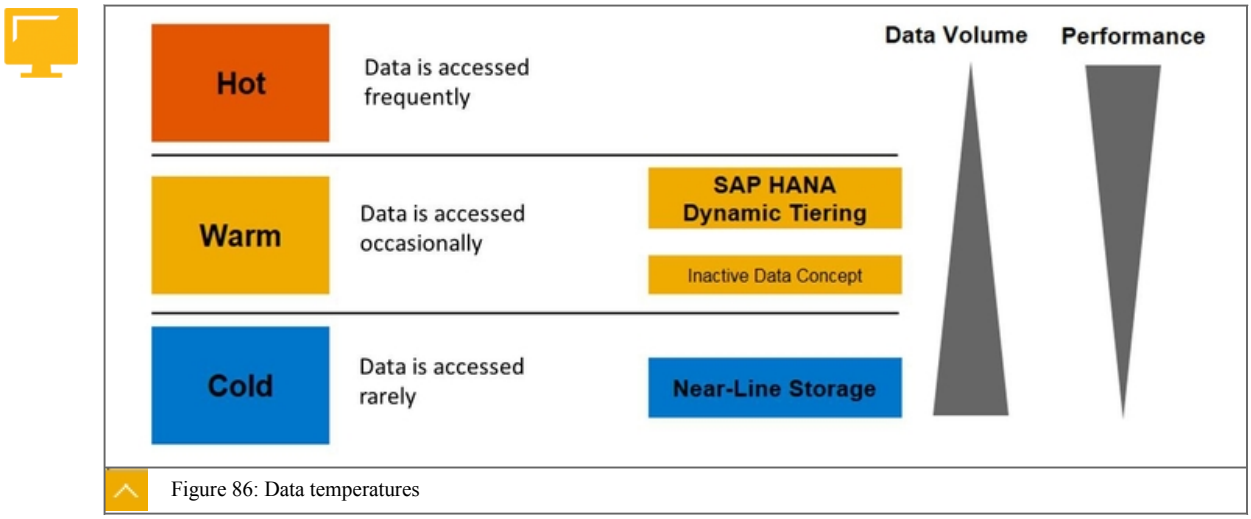


Figure 86: Data temperatures

You should size HANA memory according to hot data usage. This is an important calculation that can ensure that you get the right performance at the right price. You implement cheaper tiers such as disk and archives for the older data.

You also need to think about SLAs; that is, the performance requirement for different types of data. There is so much data acquired today that you need the think carefully about where to store this data.

Hot data is accessed frequently and/or needs very high performance. The storage for this is HANA memory.

Warm data is accessed less frequently and/or does not need high performance. The storage for this is disk, either via Extended Storage or inactive data (data that has been displaced from memory and pushed to disk).

Cold data is accessed rarely and/or does not need high performance. The storage for this is outside HANA, perhaps Near Line Store (NLS).

It is possible to classify and move data across the storage tiers automatically. Usually, it is the job of the application to control this based on usage patterns. For example in BW, you classify PSA as disk (warm) store, as this data is only needed periodically and should not occupy memory (hot).

It is also possible to control the movement of data across tiers under the control of HANA. For example, prior to a job running you can execute a command to load data to memory from disk. Then, once the job has ended, unload the data back to disk to clear out memory.

Introducing SAP Enterprise Architecture Designer

SAP Enterprise Architecture Designer is a tool that is introduced with SAP HANA 2.0. It is used to support the development of a data management solution. With this tool, you are able to develop a logical database and data management model and then turn it into a physical model.

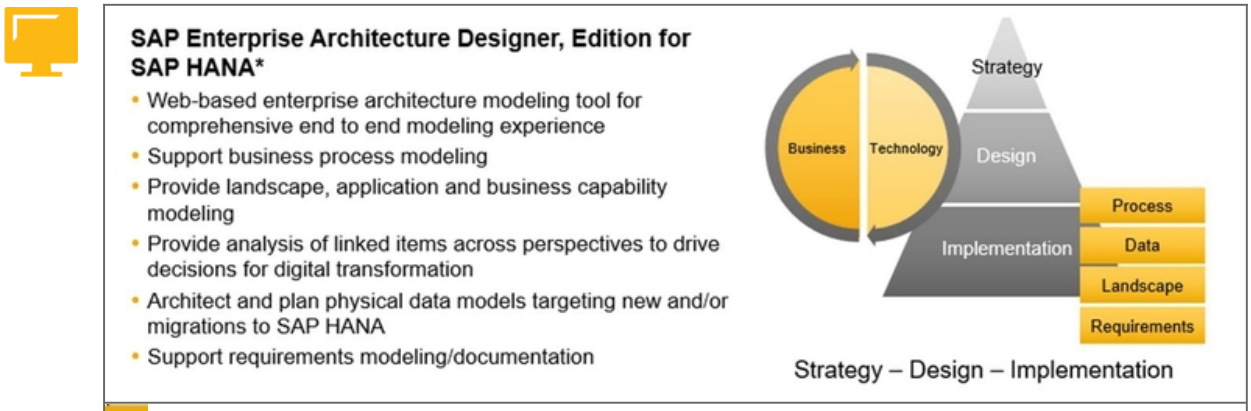


Figure 87: SAP Enterprise Architecture Designer, Edition for SAP HANA

So instead of diving into HANA to create physical schemas, tables, views, and so on, you can first develop the basic shape of the data model. You can then share this with your teams to consider the impact, and collaborate to improve the design.

Once the design is agreed, the tool can automatically generate the physical objects in SAP HANA, saving you the manual effort of doing this. This web-based tool is based on SAP PowerDesigner and it runs on SAP HANA 2.0

LESSON SUMMARY
You should now be able to:

- Describe SAP HANA data management

Unit 4
Lesson 2

Describing Data Acquisition Tools

LESSON OVERVIEW
This lesson describes the big picture of data provisioning in an SAP HANA landscape. The lesson will help the learner to understand the various scenarios for data provisioning.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe SAP HANA data acquisition solutions

Data Virtualization

Customers have to deal with complex system landscapes across different locations, storing huge amounts of data in different formats and on different platforms. Customers require a cost-efficient, easy-to-deploy solution, to get real-time data visibility across their fragmented data sources. Examples of these include operational reporting, monitoring, predictive analysis, and transactional applications. With SAP HANA, we have built-in tools that can be used to create connections to remote data sources. This means that the data can be accessed in real time as required, as if it was actually stored in an SAP HANA database.

Smart Data Access (SDA) is the name of the built-in tool set that provides an extensive catalog of adaptors to connect to remote sources. SDA can figure out on-the-fly whether a query should be pushed down to the remote data source for processing, or whether the raw data should be fetched, and the query then runs in SAP HANA. SDA always uses the approach that offers the best performance.

Motivation for Smart Data Access

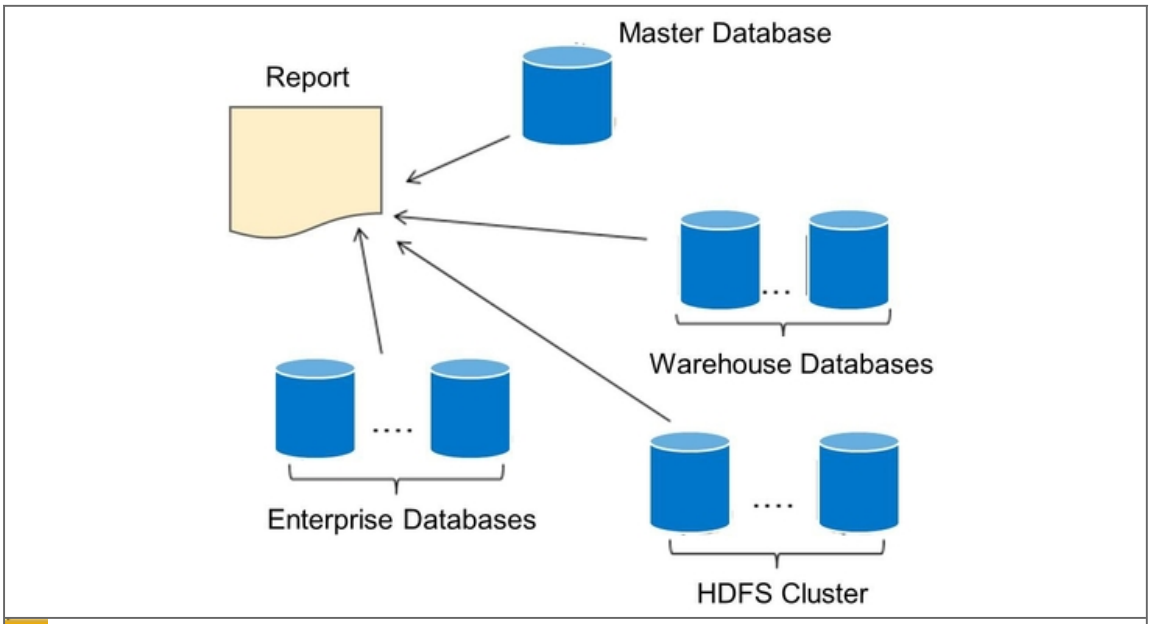


Figure 88: Motivation for Smart Data Access

To the application developer, the data appears to come from one source; that is, SAP HANA. Once the one-time connection to the remote source is established by IT, the application developers do not need to concern themselves with the technicalities of where the data is coming from.

SDA supports a modern data-federation strategy, where movement of data is minimized, and global access is enabled to data stored in local repositories.

SDA can be utilized in the following situations:

- To build new applications on SAP HANA, but access data from multiple sources without moving data into SAP HANA.
- To expose Big Data stores such as Hadoop to SAP HANA-based applications.
- To provide real-time access to archived data.
- To combine data from multiple SAP HANA-based data marts.

SAP HANA Smart Data Access: The Glue of the Platform

You can now create a fast and flexible data warehouse without expensive ETL or massive storage, security, and privacy risks. You can build big data applications with fast and secure query access to data, while minimizing unnecessary data transfers and data redundancy. You can bring social media data and critical enterprise information together, giving comprehensive insight into customer behavior and sentiment.

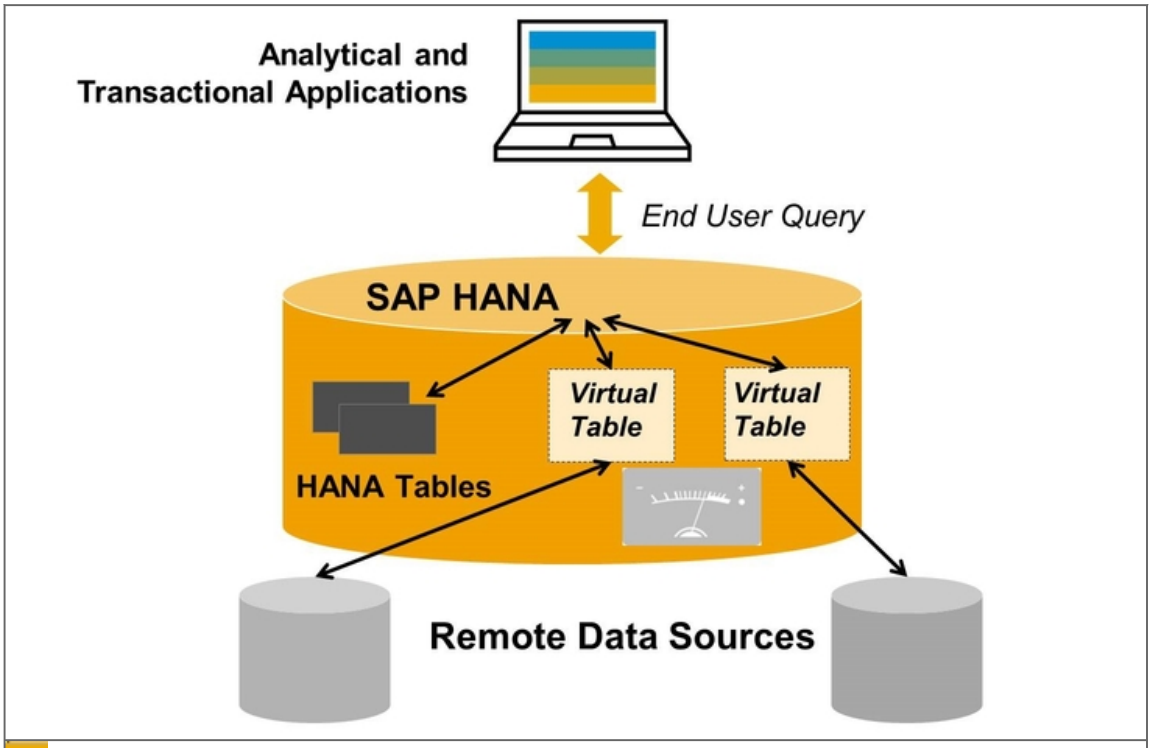


Figure 89: SAP HANA Smart Data Access: The Glue of the Platform

Benefits of SDA

SAP HANA SDA enables remote data to be accessed via SQL queries as if they were local tables in SAP HANA, without copying the data into SAP HANA. Specifically, in SAP HANA, you create virtual tables that point to remote tables or views in different data sources. Customers can then write SQL queries in SAP HANA, which can operate on these virtual tables. The virtual tables sit alongside the regular tables in the same SAP HANA database schemas. Although we are focusing here on the use of SDA to read data from remote sources, it should also be noted that SDA is able to write data to the remote source also.

Note:
A virtual table can easily be spotted in the database catalog as it has a small connector symbol added to the table icon.

The SAP HANA SQL query processor optimizes the queries that are based on virtual tables. It does this by determining if any query operations could be faster if they were pushed down to the remote source rather than processing in SAP HANA. To support this invisible decision making, SAP HANA collects statistics on the remote data sources.

The following are some of the benefits of SDA:

- Enables access to remote data just like local table
- Smart query processing pushes as much processing as possible to target data source
- Smart query processing includes query decomposition with predicated push-down, and functional compensation
- Automatic data-type translation enables remote data types to be mapped to HANA data types on the fly

- Supports data location agnostic development
- No special syntax to access heterogeneous data sources
- Provides SAP HANA to SAP HANA queries
- Supports Insert, Update, and Delete in many cases
- Calculation view support for Virtual Tables
- Delivers Generic Adapter framework to extend additional Remote Sources

If metadata changes in the remote database, it can be easily resynced with the HANA virtual table. This resyncing is done with no disruption to dependent objects (this feature was introduced with SAP HANA 2.0).

Virtual tables also cache their results so that identical queries do not have to fetch the same data again. The cached data, if unexpired, can be used again, which vastly improves performance (this feature was introduced with SAP HANA 2.0).

Note:
For the up-to-date list of adaptors, check SAP Note 1868209.

For detailed coverage of SDA, refer to the training course HA350 where we set up virtual tables.

Data Replication

Data replication typically means ensuring that data created in one system is duplicated to one or more target systems. It is usually done in real time and is managed record by record. However, replication does not always happen in real time. Replication can also take place periodically, for example, every five minutes. Periodic replication is used when it is not essential that data is always synchronized in real time.

Typically, with replication, no transformation takes place on the data as it travels to the target system so that we have an identical copy of the data in all systems. Replication involves the physical moving and loading of data, and not simply exposing the data sources as virtual tables.

The following are examples, which illustrate data replication:

- Sales orders entered into various SAP ERP systems are immediately replicated to a central data mart. This means real-time dashboards can be developed to show the live sales pool.
- A popular vending machine sends its stock information every ten minutes to the central inventory system at HQ. This can trigger replenishments when stocks are low.
- Orders are collected centrally, and simultaneously a copy of the order is routed immediately to the relevant warehouse for processing.

There are many different technical implementation approaches that support replication, ranging from the use of database logs to the use of database triggers. It is essential that the source or target system has some way of knowing that data has changed, so that replication can be kicked off.

The SAP HANA real-time replication solutions provide technologies for replicating data, in real time or batch, from any source system to the SAP HANA database. These include trigger-based data replication using **SAP LT Replication Server (SLT)** , log-based data replication

technology using **SAP Replication Server** , and session-based synchronization using **SAP HANA Remote Data Sync.**

SAP LT Replication Server (SLT)

SAP LT Replication Server (SLT) is a popular tool used by customers to ensure that data generated in an SAP Business Suite application is replicated in real time to SAP HANA. One of the main use cases for SLT is to provision data to SAP HANA for BI cases. For example, live dashboards can be kept up-to-date with real time transaction data.

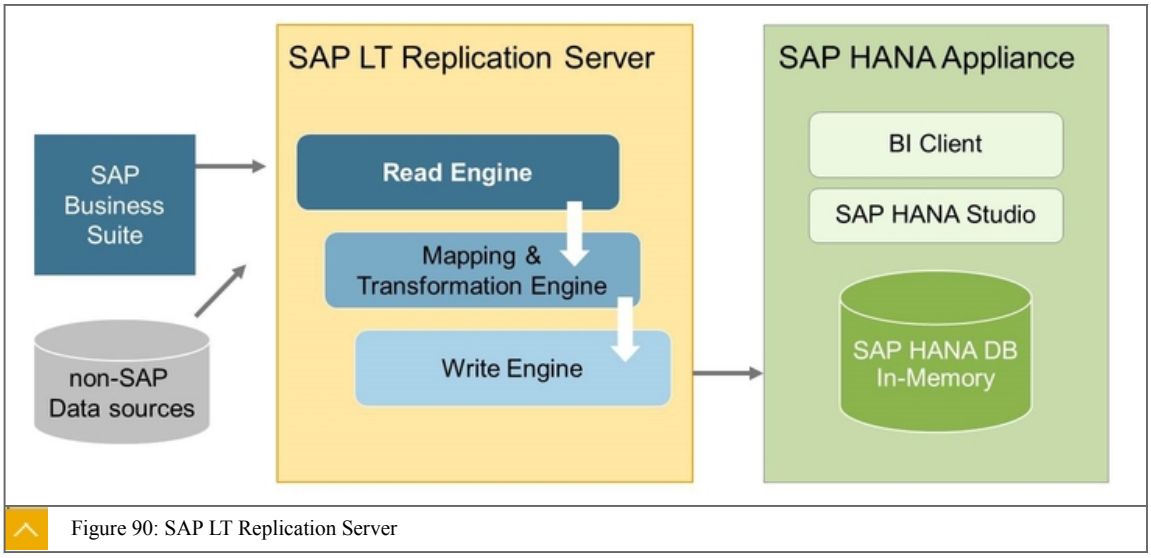


Figure 90: SAP LT Replication Server

SLT is an SAP NetWeaver ABAP-based application and it uses well known SAP technologies such as RFC and DB Connect to establish source and target connections. SLT is also used as the data transfer tool for many SAP products, such as SAP BW and SAP Accelerators. SLT plays a key role when SAP HANA Live is deployed as a side car, by managing all data replication.

SLT has been used for many years as a data transfer tool in landscape transformation scenarios (company acquisitions where data needs to be consolidated, or split). SLT predates SAP HANA. In the last few years, SLT has had many enhancements specifically aimed at its use with SAP HANA as a replication tool. Many of these enhancements help to improve the throughput of data as well as the monitoring of the data movement.

SLT is a trigger-based data provisioning tool. This means that SLT sets database triggers on any table that you want to replicate from.

When the database table is changed in any way (insert, update, or delete), the trigger is automatically fired. SLT hears the trigger and fetches the data from the source system and transfers it to SAP HANA.

SLT can perform the following types of data movement:

- Load

  This is a one-time data copy from the source system to SAP HANA. This is not replication, but a bulk copy. This tool is also used for data migration, which is typically a one-time event, so this feature is very important.

- Replication

SLT performs an initial full load of all data, and then immediately sets up a real-time replication of data from the source to the target system. This replication is trigger-based, meaning that a DB trigger is defined in the source database on each table marked for replication. Each time a data modification is done to a source table, the trigger captures this event and SLT transports this change to the target database. This is typically the most popular use of SLT, to enable feeding of real-time data to SAP HANA-based applications.

When SAP HANA tables are the target, SLT replicates data from the source system at the table level. Some data provisioning tools are able to replicate from the application level using business views (for example, BW data sources). This means that you need to know the names of the source tables you wish to replicate from.

When we think about replication we usually assume that data is moved unchanged from the source to the target. However, in some cases, you may need to apply some transformation to the data.

Although SLT is not a heavyweight data transformation tool, it is certainly possible to modify the data during the transfer process. The types of possible modifications are as follows:

- Add or remove table columns

  For example, with a wide transactional record, perhaps only a few columns are needed for reporting, or perhaps a new column to store a new calculation is needed.

- Change data type for a column

  For example, the source column is an integer type and you wish to convert this to a character string so it is able to hold more flexible values.

- Filter

  For example, you only need to replicate the orders flagged as ‘URGENT’. The filters can be set on multiple columns.

- Modify data

  For example, you wish to convert column values to align to and with other systems (e.g. you load records with country code ‘GER’ but they need to be converted to the corporate standard ‘DE’).

- Split up a table

  This is not strictly a modification, but SLT enables the distribution of data from a single source table to multiple SAP HANA target tables (and the other way around).

ABAP is used to develop the transformation logic. So, this is a crucial skill to have on any SLT project where transformations will be made.

Any transformation applied to data as it is being replicated will have an effect on the time it takes for the data to arrive at the target. For this reason, only light transformations should be implemented.

Writing data transfer rules for complex integration and cleaning can get complicated. There are better SAP data provisioning tools to use in those situations.

The administration of replication using SLT is fully integrated into the SAP HANA Studio. Here, you can choose from a number of options to stop and start data movement jobs.

SLT is a key tool used with SAP HANA Live side-by-side scenario, as well as Central Finance with S/4HANA.

Note:
You can learn more about SLT in the SAP course, SLT100 .

SAP Replication Server

SAP Replication Server is a sophisticated transactional data movement product that moves and synchronizes data across the enterprise. This is done without geographical distance limitation to meet demanding requirements in the enterprise such as guaranteed data delivery, real-time business intelligence, and zero operational downtime. SAP Replication Server facilitates this by non-intrusively handling data at the source and target database level, while ensuring high performance and transactional integrity.

SAP Replication Server is often used by organizations that need to move a lot of data in different directions in real time and ensure 100% synchronicity. You will find this solution used in many financial institutions where systems must be completely in step, in real time, with robust recovery options in case of failure.

Some of the key benefits of SAP Replication Server are as follows:

- Log-based replication process
  - Non-intrusive
  - Very high performance
- Improve recovery, resumption times and minimize downtime
  - Bidirectional replication
  - Standby DB is always available and can be used for read-only report server
- Fresh data to enable a timely decision
  - Run resource-intensive reports on reporting servers without impacting OLTP systems
  - Reduce information latency for reporting and optimize batch reporting
- Real-time data sharing and synchronization
  - Facilities decentralized business operations
  - Enables remote applications to access data locally for improved performance

One key feature of SAP Replication Server is that it relies on a log-based replication technique. The Changed Data Capture (CDC) is not done against the data volumes of the source database tables, but instead by reading directly from the database log.

Note:
A database log is a history of all actions executed by the database management system. A log is often used in the recovery of databases after a crash. When replayed, all updates to the database can be re-created.

This log-based approach reduces the workload that the replication process usually brings to the source database, thus enhancing the availability of this system.

SAP Replication Server has been enhanced recently to support replication scenarios that include SAP HANA as a source or target.

Among the latest enhancements is the ability to replicate from Business Suite applications, as SAP Replication Server is now able to handle the SAP proprietary ABAP cluster tables.

While SAP Replication Server can handle large volumes of non-disruptive replications, there are other SAP HANA replication options that can achieve satisfactory results with a much simpler landscape.

Note:
For more details on SAP Replication Server, refer to the SAP course EDB374 .

SAP HANA Remote Data Sync

SAP HANA remote data sync is a synchronization technology designed to synchronize remote databases with a central database. SAP HANA is the central database and the remote sources can be either of the following:

- SAP SQL Anywhere

  This is a powerful, highly scalable relational database that can be deployed in large or small remote applications.

- UltraLite

  This is a component within SAP SQL Anywhere used to support small footprint applications found in handheld devices such as smartphones or tablets.

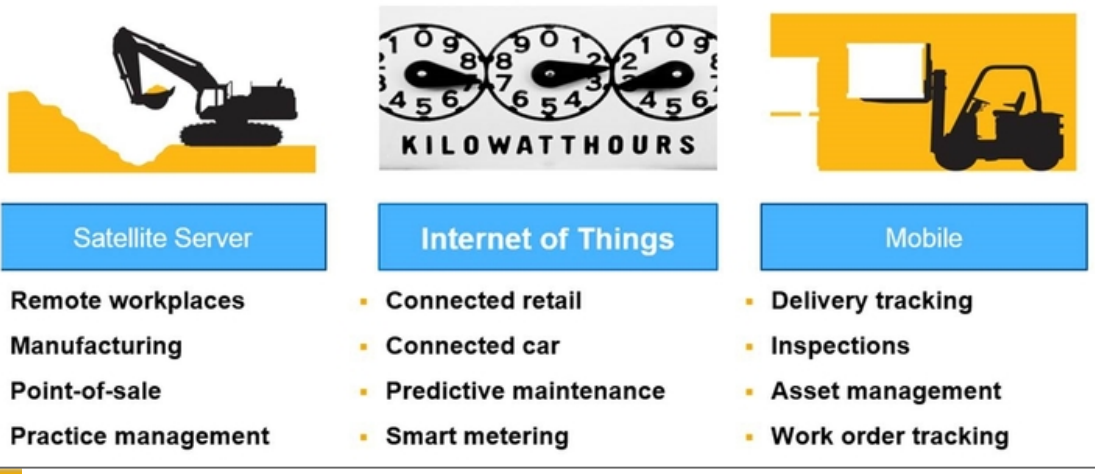


Figure 91: SAP HANA Remote Data Sync

SAP HANA remote data sync is useful when applications cannot remain continually connected to a central database due to connection problems. Example of this could include a field engineer in a remote location with a poor signal, or perhaps if the application should not be continually connected due to connection costs.

When implementing SAP HANA remote data sync, you develop occasionally connected, smart client applications. These applications sync with the central database, either periodically at set times or triggered by an event.

In all remote data sync applications, the remote data sync server is the key to the synchronization process. Synchronization typically begins when a remote data sync remote

site opens a connection to a remote data sync server. During synchronization, the remote data sync client at the remote site can upload database changes that were made to the remote database since the previous synchronization. On receiving this data, the remote data sync server updates the consolidated database, and then downloads changes from the consolidated database to the remote database.

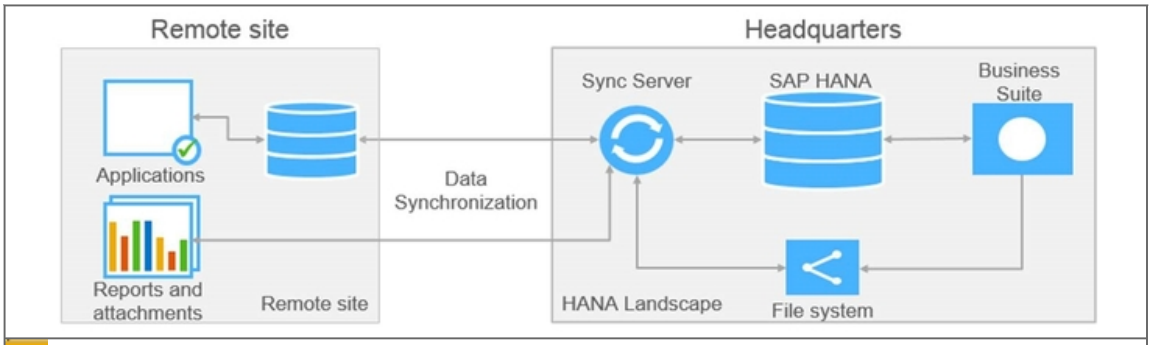


Figure 92: Remote Data Sync Architecture

One of the key advantages of SAP HANA remote data sync is that it maintains data integrity at all times. It remembers the exact sequence of updates from all remote clients. Imagine for example if 1,000 field engineers were withdrawing the same spare part, while at the same time other remote works were replenishing the same spare part. It could be easy for the stock balances to get messed up in fast, bidirectional data traffic. Fortunately, SAP HANA remote data sync handles the updates in sequence, with 100% accuracy.

Extract, Transform, Load (ETL)

Extract, Transform, Load (ETL) is the process of extracting data from source systems and applying transformations on the data, before loading to a target. This process is popular with data warehouses, such as SAP BW where there are many data sources that need merging. SAP HANA has two key tools that can be used in the data provisioning scenario around ETL. These are SDI/SDQ and SAP Data Services.

Smart Data Integration (SDI)/Smart Data Quality (SDQ)

There are many options to choose from when considering data provisioning to SAP HANA. Most of these options require the installation and setup of additional software and hardware components, which sit between the data source and SAP HANA. These components cover a broad range of capabilities, including extracting data, combining sources, and cleansing, loading, or exposing the data to SAP HANA.

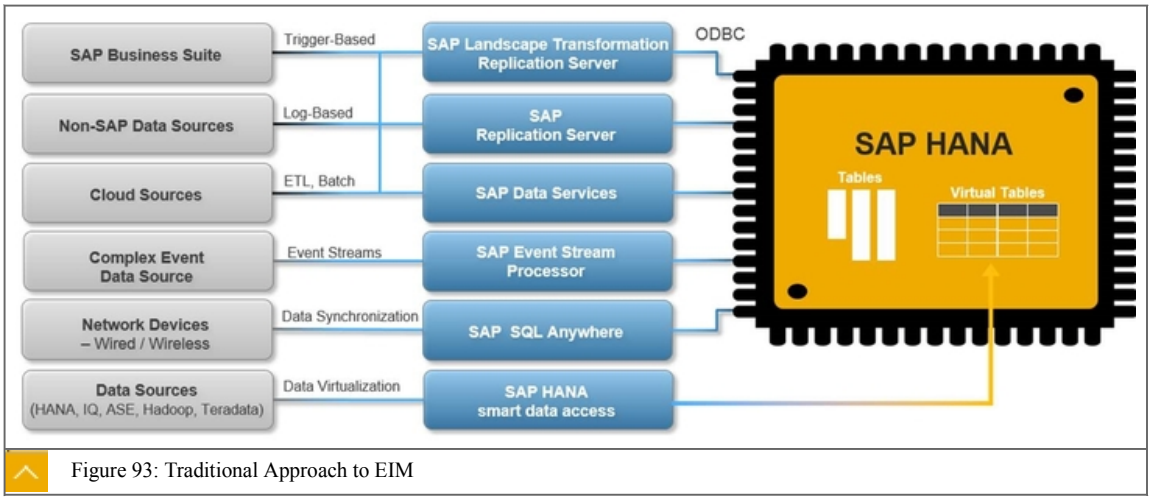


Figure 93: Traditional Approach to EIM

The inclusion of an additional data provisioning tier adds to the complexity of the overall solution.

A Simplified Approach to Data Integration

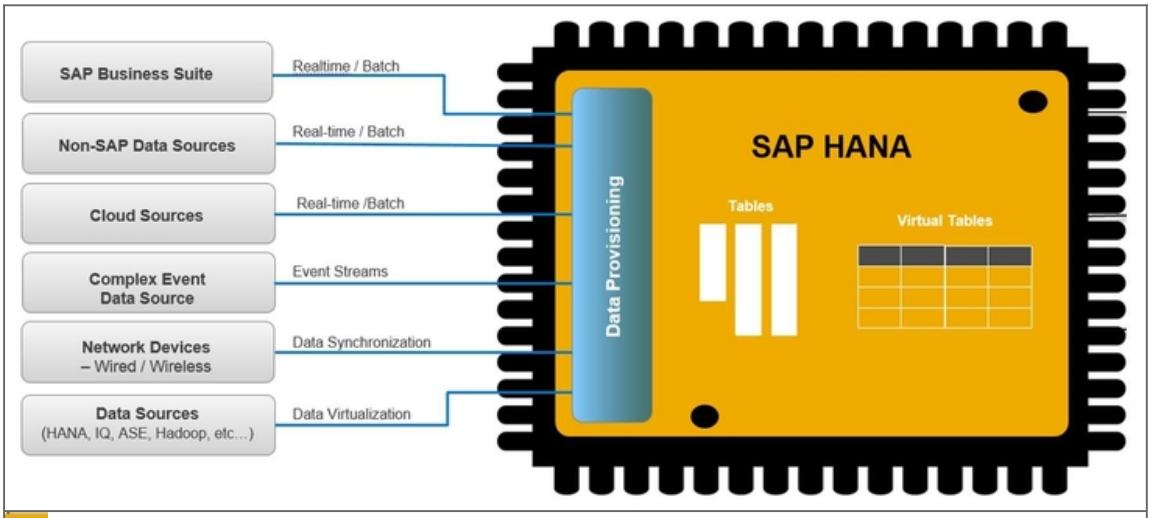


Figure 94: Simplified Approach to EIM

SAP HANA has its own built-in ETL capabilities. We call this Enterprise Information Management (EIM). The two components of EIM are Smart Data Integration (SDI) and Smart Data Quality (SDQ). This means that no additional tools and associated hardware are required, as everything you need is provided with SAP HANA.

With EIM, we have removed the external data provisioning tier. Running data provisioning tasks inside SAP HANA also means that we take advantage of the high performance, in-memory processing for data acquisition tasks.

Components of EIM

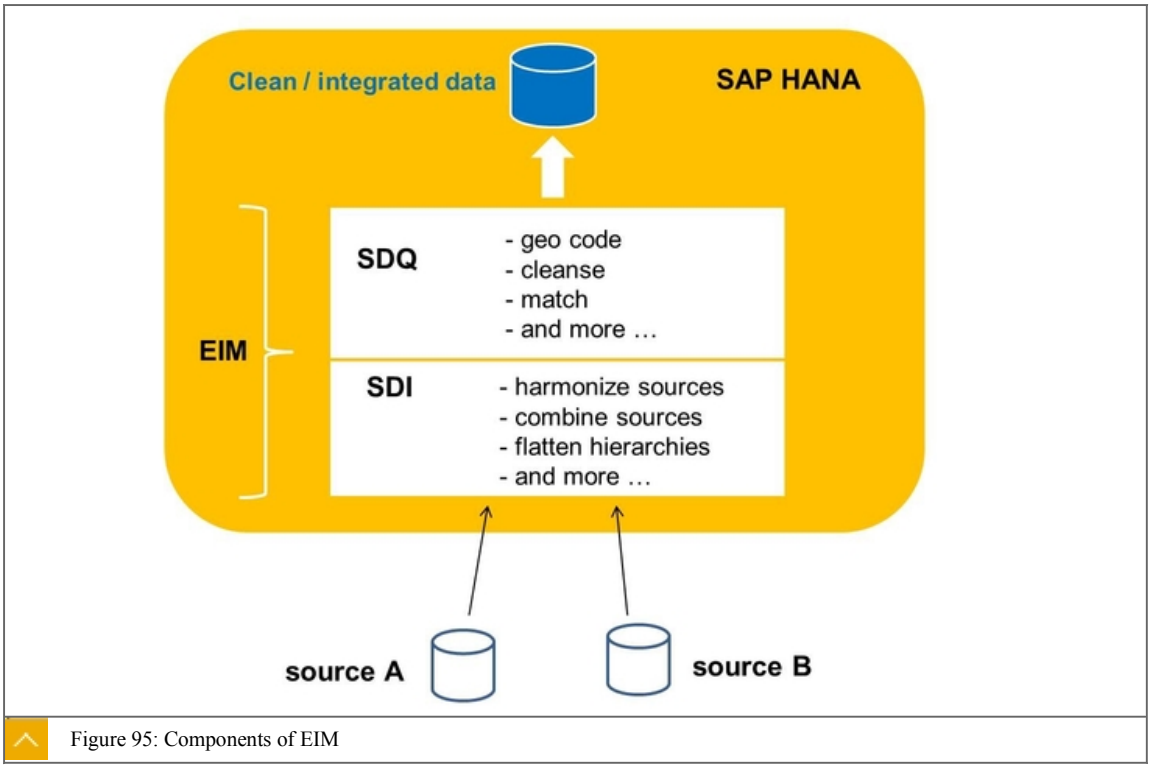


Figure 95: Components of EIM

The following are the components of EIM:

- Smart Data Integration (SDI)

  Functions for acquiring and integrating data from multiple sources

- Smart Data Quality (SDQ)

  Functions for improving data quality

Although we see two components, SDI and SDQ, do not think of these as two separate products. When building any data provisioning job, the developer is able to freely include any of the capabilities from either component. SDI is the key component that takes care of data acquisition and integration, whereas SDQ can add additional steps to the job to enhance and clean the data. SDQ relies on the basic features of SDI to get the data moving.

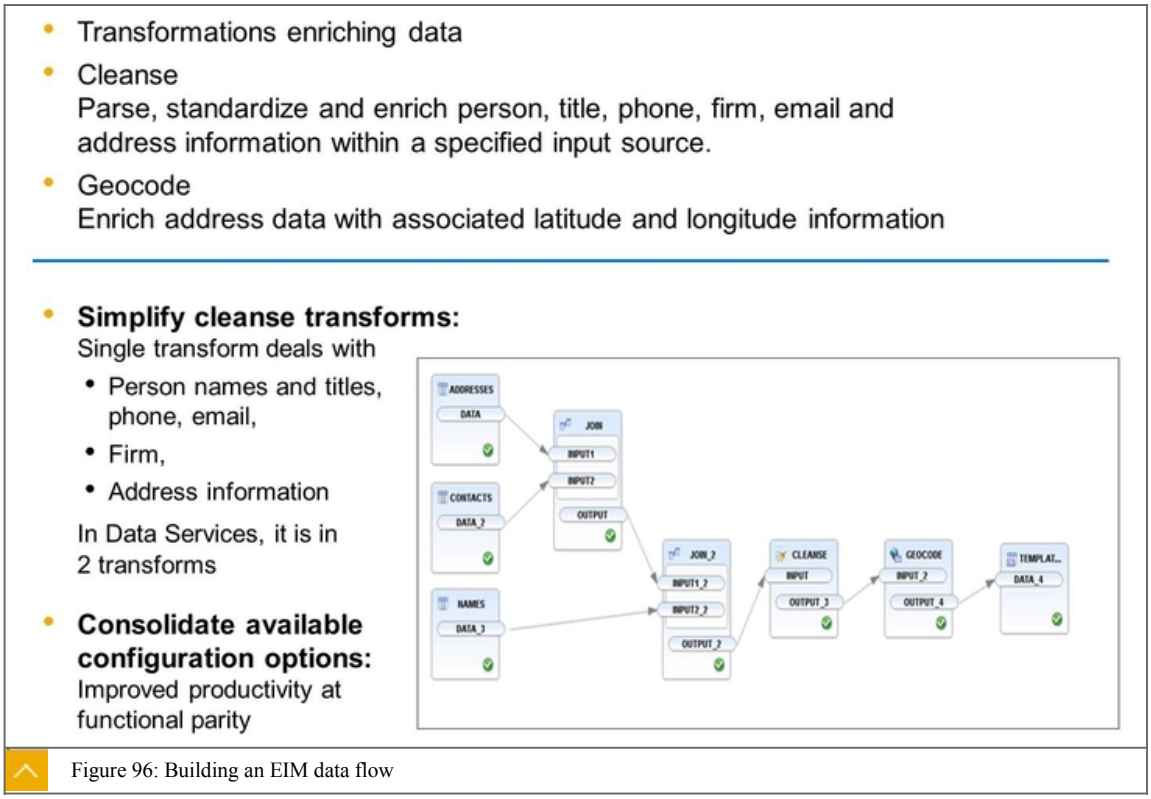


Figure 96: Building an EIM data flow

EIM jobs are created in SAP HANA using flowgraphs. Flowgraphs are graphical representations of a data provisioning job. They contain a sequence of nodes and flow lines that represent the steps in the flow. Developers create jobs by dragging and dropping the nodes to a canvas to create the flowgraph.

SAP Data Services

SAP Data Services provides the capability to extract, transform, and load (ETL) data from any source and to any target. This includes SAP HANA as both a source or target. SAP Data Services has been around for many years and is deeply embedded in the distributed landscapes of many customers. So for those customers it is good news that SAP HANA can also be integrated as a data source or target in a mixed landscape.

SAP Data Services is SAP's most powerful multi-purpose ETL platform. It does not rely on SAP HANA and operates on its own server. It provides very sophisticated data integration and harmonization features, as well as country-specific data cleansing tools.

SAP Data Services can pass and exchange metadata with SAP HANA. For example, the output structures of the Data Services jobs can automatically create SAP HANA tables. Or, SAP HANA tables and views can be automatically exposed to Data Services jobs for input.

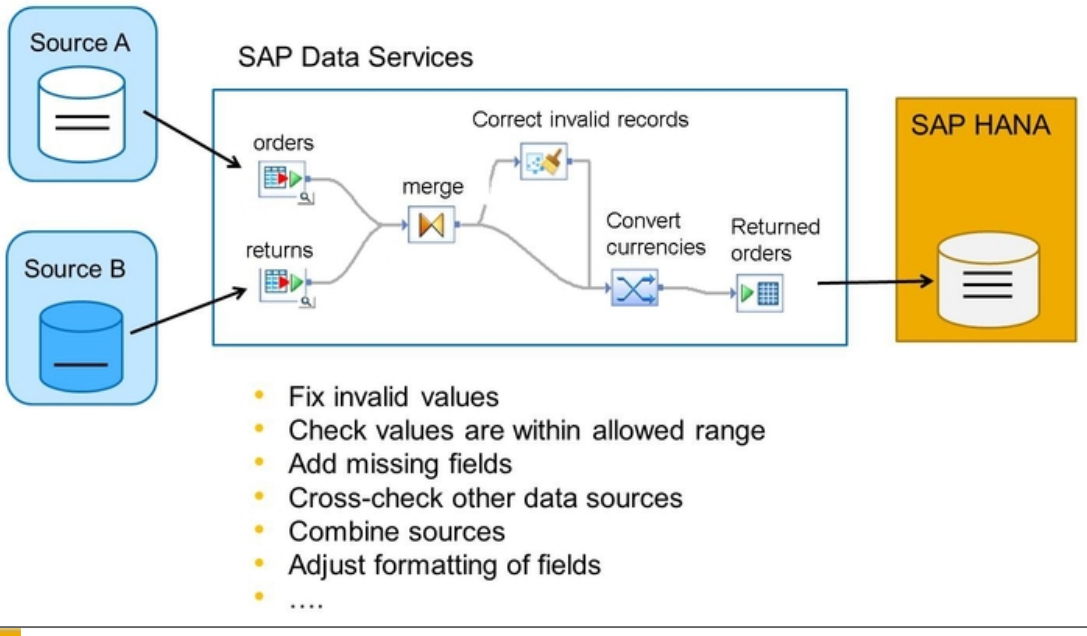


Figure 97: SAP Data Services

Core Capabilities of SAP Data Services

SAP Data Services usually processes all data in its own engines and sends the output to the target systems. When SAP Data Services is used with SAP HANA, a significant amount of the data processing is pushed to SAP HANA, to ensure ETL jobs run as fast as possible. SAP have enhanced Data Services recently to provide tighter integration with SAP HANA, so data flow jobs are optimized.

Although the core capabilities of SAP Data Services are also available in SDI/SDQ, SAP Data Services continues to provide many more data integration and quality transforms that are not yet available in SAP HANA SDI/SDQ. SAP Data Services excels at managing complex delta loading to data warehouses with auto-recovery mechanisms built-in to restart jobs if they fail. SAP Data Services can also quarantine data that does not pass quality checks for more intensive processing.

SAP Data Services is a family of tools and a key component is SAP Information Steward. This tool provides extensive data quality and data profiling dashboards so business users can monitor and measure data quality. The tool also exposes the data cleansing rules to business users who can create and adjust cleaning rules without the need for IT involvement. Enterprise-wide data lineage is also a capability of Information Steward so that the origins of data can be traced from reports.

Over time, we may find some or all of these capabilities appearing in SAP HANA. However, for now SAP Data Services remains a good choice for many customers who need a fully loaded ETL solution with sophisticated features for a complex data landscape.

Unit 4
Exercise 8

Extract, Transform, and Load Data Using SDI

Each week, HR provide a list of recently joined employees in a simple flat file that they place in a network Windows folder. Sometimes you need to manually add records to the file that have been missed. You would like to load the records to a SAP HANA table and in order to determine the type of uniform each employee will need, you would like to enrich the records by having SAP HANA determine the gender.

1. Examine the contents of the source flat file consultants.csv using Notepad which is located in Favorites → HA100 .
2. Add a new record to the end of the file making up a new employee name, as follows:
   `0007,any first name,any second name,70`
   Don’t forget the three separator commas, and also don't forget to add a new first and second name.
3. In the Web IDE, create a new flowgraph with the name **`load_employees_##`**.
4. Add a node to the flowgraph to define a data source based on your virtual table Consultants_## .
5. Add a node after the data source node to cleanse the source data.
6. Connect the Data Source node with the Cleanse node.
7. Configure the Cleanse node to generate an extra column that identifies the gender of each employee.
8. Add a node to define a data target.
9. Connect the Cleanse node with the Data Target node.
10. Configure the Data Target to use a template table with the name Employees_Gender_## .
11. Save and build the flowgraph.
12. Execute the flowgraph.
13. Check the results in the newly created target table Employees_Gender_## .

Note:
If you want to re-run the flowgraph, you will need to add the setting Truncate Table in the Settings of the Data Target node, save, and rebuild the flowgraph. Otherwise the load will fail with a key violation error due to duplicate keys.

Unit 4
Solution 8

Extract, Transform, and Load Data Using SDI

Each week, HR provide a list of recently joined employees in a simple flat file that they place in a network Windows folder. Sometimes you need to manually add records to the file that have been missed. You would like to load the records to a SAP HANA table and in order to determine the type of uniform each employee will need, you would like to enrich the records by having SAP HANA determine the gender.

1. Examine the contents of the source flat file consultants.csv using Notepad which is located in Favorites → HA100 .
   a) Expand the folder Favorites → HA100 . Right-click on the file consultants.csv and choose Notepad to display the contents. (Do not simply double-click.)
2. Add a new record to the end of the file making up a new employee name, as follows:
   `0007,any first name,any second name,70`
   Don’t forget the three separator commas, and also don't forget to add a new first and second name.
   a) Type **`0007,any first name,any second name,70`** at the end of the file. Don’t forget to add a first and second name.
   b) Save the file using File → Save .
   c) Close the file.
3. In the Web IDE, create a new flowgraph with the name **`load_employees_##`**.
   a) Expand the folders HDB → src .
   b) Right-click on the folder exercises and chose the menu option New → Flowgraph .
   c) At the prompt, enter the name **`load_employees_##`** and choose OK.
4. Add a node to the flowgraph to define a data source based on your virtual table Consultants_## .
   a) Click on the large blue + button and from the list that appears, choose Data Source . Click anywhere to the left of the canvas to drop the node.
   b) In the Data Source node, choose the configure symbol (looks like a cog) at the bottom of the node.
   c) Choose the button HANA Object so the Data Source selector appears.
   d) In the search field enter **`cons`** so that the virtual table Consultants_## appears in the list below.
   e) Highlight the row that contains the name of your virtual table Consultants_## and choose Finish .
   f) Choose Apply to save and close the node.
5. Add a node after the data source node to cleanse the source data.

a) Click on the large blue + button and from the provided list, choose Cleanse and then click anywhere in the center of the canvas to drop the node.

6. Connect the Data Source node with the Cleanse node.

a) Drag a line from the output port on the Data Source node to the input port IN on the Cleanse node.

7. Configure the Cleanse node to generate an extra column that identifies the gender of each employee.

a) In the Cleanse node, choose the configure symbol (looks like a cog) at the bottom of the node.

b) Choose the Edit Defaults button and from the menu choose Edit Content Types to observe the suggested content types that SAP HANA has assigned to each column of the source data.

c) Assign the column NAME to the data content type Unknown as this column is the source file name, not a person's name.

d) Observe how the input columns FirstName and LastName have automatically been assigned the correct content types.

e) Choose Save .

f) Choose the circular button Cleansed Output and choose the button Customize Manually .

g) De-select the two columns that have been automatically checked. You will not use these.

h) Navigate to the component Person → Person Extended and then select the option Gender (MALE_STRONG) and choose Save .

i) Choose the circular button Output Summary and review the final output columns that now include the generated column for gender.

j) Choose the Apply button to save the node and return to the canvas.

8. Add a node to define a data target.

a) Click on the large blue + button and from the provided list, choose Data Target and then click anywhere to the right of the canvas to drop the node.

9. Connect the Cleanse node with the Data Target node.

a) Drag a line from the output port on the Cleanse node to the input port on the Data Target node.

10. Configure the Data Target to use a template table with the name Employees_Gender_## .

a) In the Data Target node, click on the configure symbol (looks like a cog) at the bottom of the node.

b) Choose the Template Table button and in the Object Name field enter **`Employees_Gender_##`**.

c) Choose Apply.

11. Save and build the flowgraph.

a) Choose Save .

b) Chose the menu option Build → Build Selected Files and check the log to ensure the flowgraph built successfully.

12. Execute the flowgraph.

a) At the top of the flowgraph canvas you will see a short toolbar. Click the icon to the right that looks like a triangle pointing to the right. (When you hover over it you will see the word Execute).

Caution:
On the screen you will see two icons that use the same triangle symbol.
The one in the top main toolbar is used to Run XSA applications and not to Execute a flowgraph.

b) At the confirmation prompt choose OK.

c) If it's not already open, you should open the Console to check the log. (The load takes about 3 seconds).

13. Check the results in the newly created target table Employees_Gender_## .

a) Switch to the Database Explorer view.

b) Expand the entry My HA100 Container → Tables .

c) In the lower pane, right-click on the table Employees_Gender_## and choose Open Data and you should see the employee records. Each has a gender assigned.

Note:
If you want to re-run the flowgraph, you will need to add the setting Truncate Table in the Settings of the Data Target node, save, and rebuild the flowgraph.
Otherwise the load will fail with a key violation error due to duplicate keys.

Data Streaming

Data streaming is the transfer and processing of continuous data from a source to a target. Data streaming often involves high-speed, high-volume data transfers from multiple streams in parallel. Sources of streaming data can range from simple sensors to complex business systems.

In today's highly connected digital world, data streaming is an essential enabler of real-time information, to feed applications and dashboards. The opportunities for the development of innovate applications are enormous.

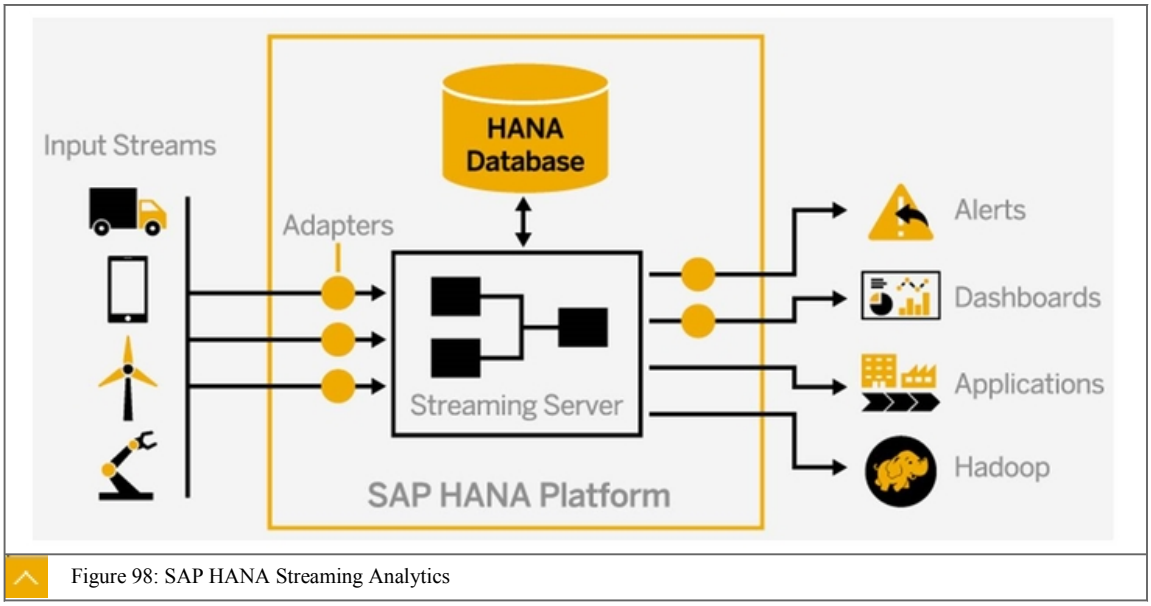


Figure 98: SAP HANA Streaming Analytics

Enterprises today are flooded with streams of notifications as things happen. Individual events may not be significant by themselves, which makes it difficult to discern when consequential events do occur. You could have thousands of sensors reporting statuses every few seconds, and most of that information is uninteresting. However, when something is starting to go wrong, you want to know as soon as possible, so that you can act before a small trend becomes a significant problem.

Data streams contribute significantly to the size of the world's digital data, known as Big Data.

SAP HANA Streaming Analytics

SAP HANA Streaming Analytics has its roots in an SAP product called Event Stream Processor (ESP), which was originally developed by Sybase. When SAP introduced data streaming to SAP HANA, it was called Smart Data Streaming (SDS). It has now been re-branded as **SAP HANA Streaming Analytics** .

The streaming analytics server runs as a separate server in the SAP HANA landscape, separate from, but interacting with, the SAP HANA database.

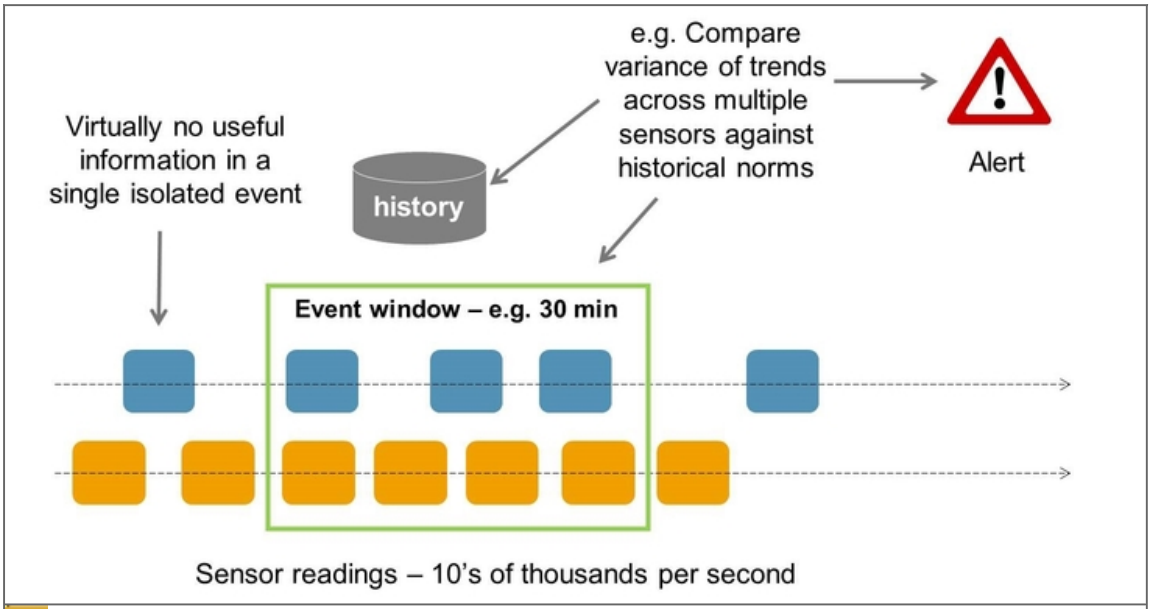


Figure 99: SAP HANA Streaming Analytics

With SAP HANA Streaming Analytics, you can capture data (millions of events per second) arriving continuously from devices and applications, and act on this new information as soon as it arrives. This enables you to react in real time using alerts, notifications, and immediate responses to continually-changing conditions. Often a single stream offers little useful information but when we combine multiple streams triggered by different events, we can begin to develop real insight.

Note:
SAP HANA Streaming Analytics is also known as SAP HANA Smart Data Streaming (SDS) and is covered in more detail in the SAP course, HA350 .

Flat File Loading

One of the most popular ways data is presented for loading to SAP HANA is through a flat file. There are multiple methods to loading flat files into SAP HANA. One of the quickest and simplest ways to load data to SAP HANA is by using the flat file import wizard in the SAP HANA Studio. The wizard supports either csv files or Excel workbooks.

Using the SAP HANA Studio Import Wizard

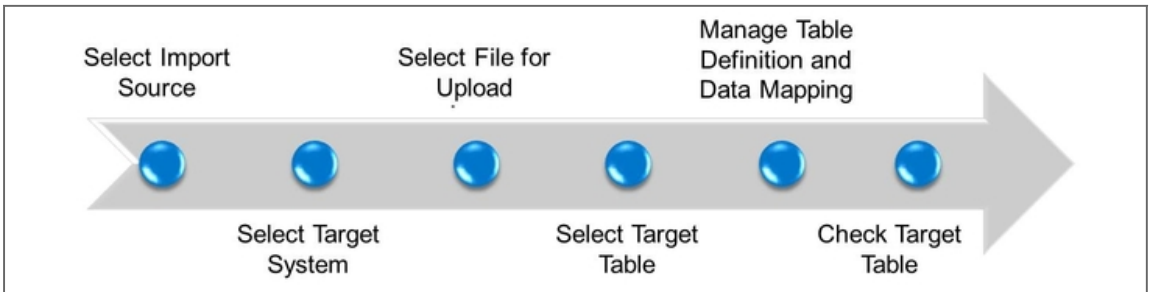


Figure 100: Process Flow: Uploading Data from Flat Files

The process to load a flat file to SAP HANA using SAP HANA Studio is simple and is guided by a wizard. The process is as follows:

1. From the SAP HANA Studio Import menu, choose the option to import data from a local file.
2. Choose the target SAP HANA database.
3. Choose the flat file to import.
4. Choose the existing SAP HANA target table to load to, or choose to automatically create a new table.
5. Confirm or adjust the suggested field mapping and column types.
6. Preview the output and if you are happy, execute the load.

Flat File Import Wizard — What You Can and Can't Do

To upload data from flat files, SAP HANA Studio flat file import wizard offers the following features:

- You can load to an existing SAP HANA table or have the wizard create the table for you.
- If the target table does not exist in SAP HANA database, the wizard creates the empty table automatically based on whatever metadata can be supplied or determined from the flat file.
- The wizard suggests the column name and data type for the new tables. It is possible to edit these suggestions.
- The new table always has a 1:1 mapping between the file and table columns. You can't split columns up or concatenate.
- You can add extra empty columns or remove columns from the source file that you don't need.
- You cannot define filters. All records are loaded.
- When loading new data to an existing table, it is appended to any existing data. Existing records are never overwritten — so make sure the new records don't create key violations.
- When loading data to an existing table you cannot overwrite the target table column names or change the data types and lengths.
- The supported file types are .csv, .xls, and .xlsx.

Big Data Integration

Many organizations have embraced Big Data, collecting and storing staggering amounts of data of all types sourced from sensors, web logs, social media traffic, communication logs, and more. Unlike traditional business data, Big Data is usually stored in an unstructured way and so it is difficult for us to define detailed semantics on this data so that it can easily be integrated with existing analytics. Organizations usually have to implement additional specialized tooling on top of their Big Data in order to add a semantic layer (to add meaning and structure to the data) and also provide query capabilities. These tools can be complex and are often used by only a small number of highly trained analysts. This often means that Big Data analysis becomes siloed from mainstream BI.

We accept that Big Data and enterprise data are typically stored separately. However, the analysis should not be separated, and business analysts should be able to consume any data and not be concerned over whether the data is Big or not.

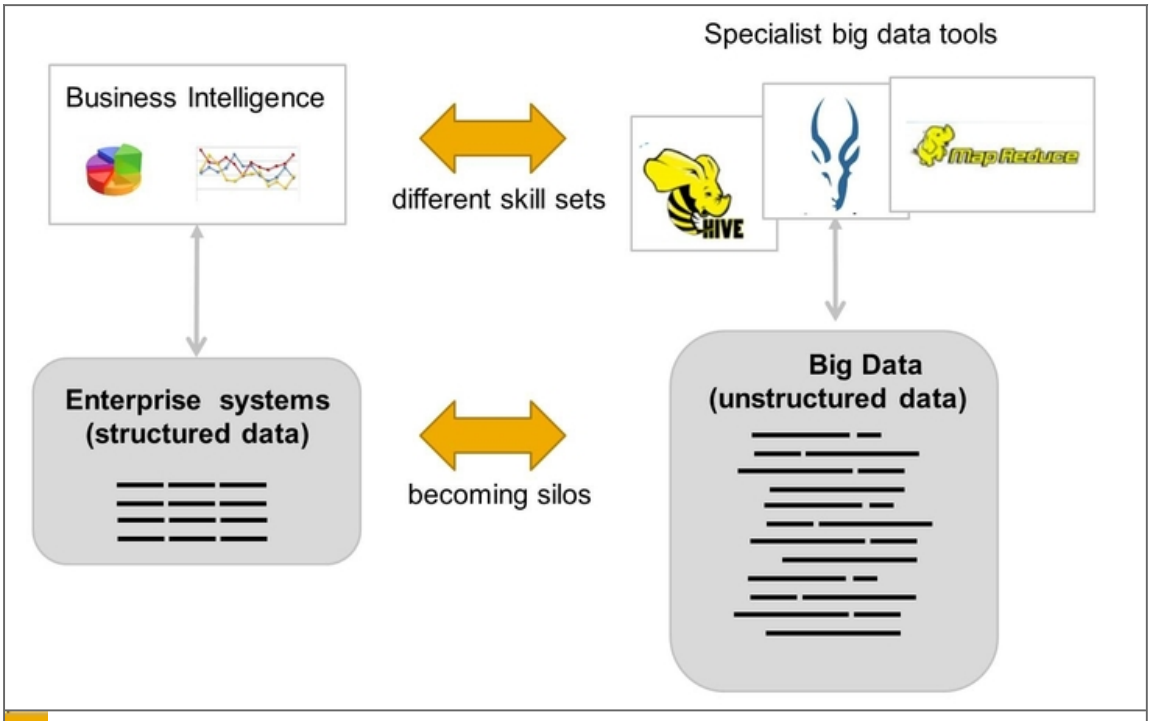


Figure 101: Big Data and enterprise data are separate

SAP Vora enables analysts to consume Big Data and enterprise data as one, using their favorite drill-down, slice and dice, query tools.

Note:
SAP Vora was formerly known as SAP HANA Vora. SAP Vora has never depended on SAP HANA and can be used separately to integrate Big Data with any applications whether SAP HANA is in the landscape or not. The name change helps to remove this incorrectly assumed dependency with SAP HANA.

Let's describe SAP Vora from a technical perspective. First, a Big Data framework consists of a data storage component. The most popular Big Data storage solution is Apache Hadoop, usually referred to simply as Hadoop. It should be noted that there are other Big Data providers and SAP Vora will work with more and more providers including Microsoft Azure Data Lake.

Hadoop provides massive data storage capabilities across cheap, everyday servers that can easily be scaled to provide staggering storage capacity. However, Hadoop does not provide the data processing capabilities and that is where Apache Spark comes in.

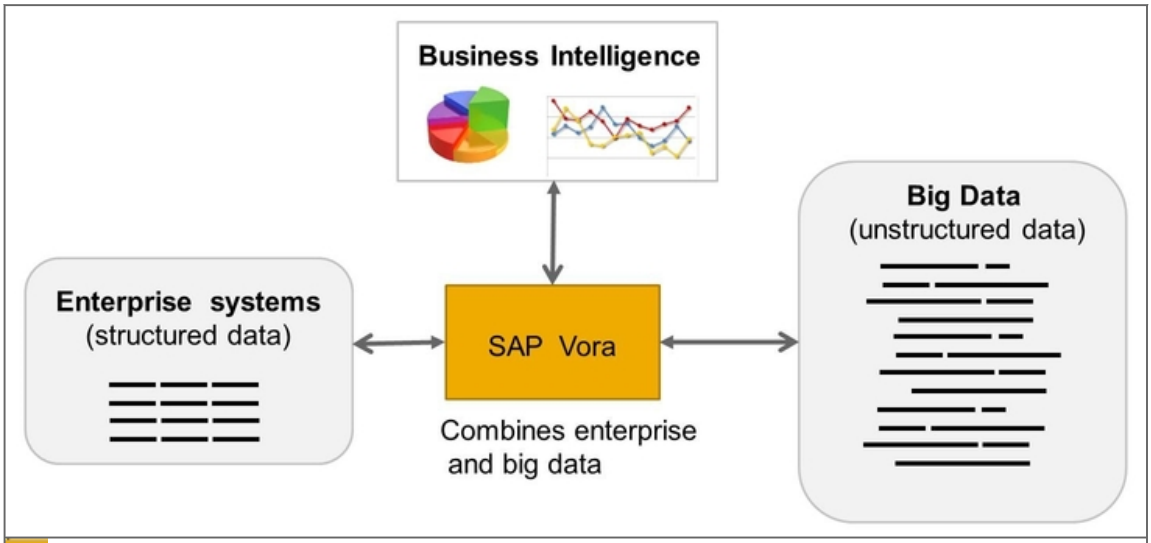


Figure 102: Bringing Big Data and Enterprise Data Together

Apache Spark (referred to simply as Spark), provides the query processing on top of Hadoop. However, Apache Spark is not able to provide the complex OLAP-type analysis features for Big Data data that most business analysts require. It is also not able to integrate enterprise data and Big Data. This is where SAP Vora comes in.

SAP Vora is an in-memory query engine that plugs into Apache Spark. It enhances Apache Spark to include hierarchies, including time-dependent hierarchies, OLAP-style dimensional drill-down/across, unit-of-measure conversion, currency conversion, and many more OLAP capabilities. SAP Vora also allows you to create precompiled queries that are ready to go, to enable fast execution.

The latest version of SAP Vora also includes support for other types of Big Data, not just Hadoop.

From an SAP HANA perspective, adding SAP Vora to the landscape brings Big Data analytics to an already powerful set of data modeling capabilities built using calculation views.

Using SAP HANA and SAP Vora you can perform the ultimate root-cause analysis by drilling down from the high-level business dashboard to the most atomic piece of individual data in a log. This provides a better understanding of the business context and helps to provide complete insight in one work flow.

Note:
When exploring SAP Vora you may come across **Altiscale** . Altiscale was one of the first companies to offer Hadoop as-a-service in the cloud, and was acquired by SAP in 2016. Altiscale is now a key part of SAP Cloud Platform Big Data Services.

The following are a list of possible SAP HANA Vora Use Cases:

1. **360-degree customer marketing** — Combine customer data with unstructured data from social media, e-mail, website activity, and discussion forums to better target sales and marketing.
2. **Cell phone service improvement** — Analyze instances of poor cellular service, such as dropped calls or poor audio, by drilling from billing data to detailed call log data.

3. **Fraud detection** — Detect anomalies and rogue trades by analyzing historical trends and current data simultaneously.
4. **Airline maintenance planning** — Combine aircraft sensor data, collected in-flight, with flight schedule and staffing data to optimally plan aircraft downtime.
5. **Targeted network maintenance and upgrades** — Analyze the impact of cable network congestion on churn, and identify which network upgrades will produce greater incremental revenue.

One of the key aims of SAP Vora is to support push-down of the query to the source of the Big Data. It is possible to simply use SDA to expose Big Data to SAP HANA directly, but this does not enable push-down of the query. In that case the data is extracted from Hadoop and presented to SAP HANA for processing. This does not encourage good performance.

LESSON SUMMARY
You should now be able to:

- Describe SAP HANA data acquisition solutions

Unit 4

Learning Assessment

1. In data temperature management, cool data is stored in HANA Extended Storage.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. What is SAP Enterprise Architecture Designer?

   Choose the correct answer.

   - ☐ **A** A tool to support sizing of an SAP HANA solution
   - ☐ **B** A tool used to define a data loading job
   - ☐ **C** A tool to support the build of a logical data model
   - ☐ **D** A tool used to simulate a work load on different HANA hardware configurations

3. What are typical characteristics of SAP HANA data replication?

   Choose the correct answers.

   - ☐ **A** Virtualization
   - ☐ **B** Duplicate data
   - ☐ **C** Real-time
   - ☐ **D** Streaming

4. SAP SLT can be used for real-time replication from SAP and non-SAP sources.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

5. What are components of SAP HANA EIM?

   Choose the correct answers.

   - ☐ **A** SDI
   - ☐ **B** SDA
   - ☐ **C** SDS
   - ☐ **D** SDQ

6. What are features of SDA?

   Choose the correct answers.

   - ☐ **A** Automatic data type translation
   - ☐ **B** Data cleansing
   - ☐ **C** Evaluation and execution of push-down possibilities to remote sources
   - ☐ **D** Merging data from multiple sources

7. Which of the following are correct statements relating to the flat file import function of SAP HANA Studio?

   Choose the correct answers.

   - ☐ **A** The new table always has a 1:1 mapping between the file and table columns.
   - ☐ **B** The supported file types for upload are .csv, and .txt.
   - ☐ **C** When loading new data in a table that already contains data, the new data is appended to the existing data.
   - ☐ **D** Renaming of columns and changing data types is allowed when loading data to new HANA tables.

Unit 4

Learning Assessment - Answers

1. In data temperature management, cool data is stored in HANA Extended Storage.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! Cool data is stored on NLS or HDFS, or any database outside HANA; Extended Storage holds the warm data. Refer to HA100, Unit 4, Data Management with SAP HANA, for details.

2. What is SAP Enterprise Architecture Designer?

   Choose the correct answer.

   [ ] **A** A tool to support sizing of an SAP HANA solution

   [ ] **B** A tool used to define a data loading job

   [X] **C** A tool to support the build of a logical data model

   [ ] **D** A tool used to simulate a work load on different HANA hardware configurations

   You are correct! SAP Enterprise Architecture Designer is a tool used to design and build a logical data model, and then, optionally, generate a corresponding physical data model : Refer to HA100 Unit 4, Data Management with SAP HANA, for details.

3. What are typical characteristics of SAP HANA data replication?

   Choose the correct answers.

   [ ] **A** Virtualization

   [X] **B** Duplicate data

   [X] **C** Real-time

   [ ] **D** Streaming

   You are correct! Data replication with SAP HANA is typically the duplication of data in real time with no changes to data. Refer to HA100, Unit 4, Data Management with SAP HANA , for details.

4. SAP SLT can be used for real-time replication from SAP and non-SAP sources.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   You are correct! You can use SAP SLT for real-time replication from SAP and non SAP sources. Refer to HA100, Unit 4, Data Management with SAP HANA , for details.

5. What are components of SAP HANA EIM?

   Choose the correct answers.

   [X] **A** SDI

   [ ] **B** SDA

   [ ] **C** SDS

   [X] **D** SDQ

   You are correct! SDS (for streaming) and SDA (for virtualizing) are not part of EIM. Refer to HA100, Unit 4, Data Management with SAP HANA , for details.

6. What are features of SDA?

   Choose the correct answers.

   [X] **A** Automatic data type translation

   [ ] **B** Data cleansing

   [X] **C** Evaluation and execution of push-down possibilities to remote sources

   [ ] **D** Merging data from multiple sources

   You are correct! SDA does not support cleansing or merging of multiple data sources. Refer to HA100, Unit 4, Data Management with SAP HANA , for details.

7. Which of the following are correct statements relating to the flat file import function of SAP HANA Studio?

   Choose the correct answers.

   [X] **A** The new table always has a 1:1 mapping between the file and table columns.

   [ ] **B** The supported file types for upload are .csv, and .txt.

   [X] **C** When loading new data in a table that already contains data, the new data is appended to the existing data.

   [X] **D** Renaming of columns and changing data types is allowed when loading data to new HANA tables.

   You are correct! You cannot transform data during the load, but you can ignore fields and even add empty ones. Supported file types are CSV and XLS. Data is never overwritten. Renaming columns and changing data types is not allowed when loading to existing tables. Refer to HA100, Unit 4, Data Management with SAP HANA , for details.

UNIT 5 Powering Applications with SAP HANA

UNIT OBJECTIVES

- Describe the types of application you can run on SAP HANA
- Run enterprise suites on SAP HANA
- Connect SAP Business Intelligence Tools to SAP HANA
- Outline how SAP BW leverages SAP HANA
- Describe SAP HANA Data Warehouse Foundation
- Describe the basics of native HANA applications

Unit 5
Lesson 1

Describing the Types of Applications you can run on SAP HANA

LESSON OVERVIEW
This lesson covers running applications on SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe the types of application you can run on SAP HANA

Applications that can run on SAP HANA

The key role of SAP HANA is to provide platform services to any type of application. The key services include not only database services, but also application processing services and data management services.

There is no limit to the type of application that can be run on SAP HANA. SAP HANA is built with open, industry-standard connection technologies. This means we leverage well-known interfaces, such as ODBC, JDBC, oData, HTTP, Web Services, and so on.

In this section we look at some examples of applications, both SAP and non-SAP, that can be run on SAP HANA.

Applications that run on SAP HANA: Examples

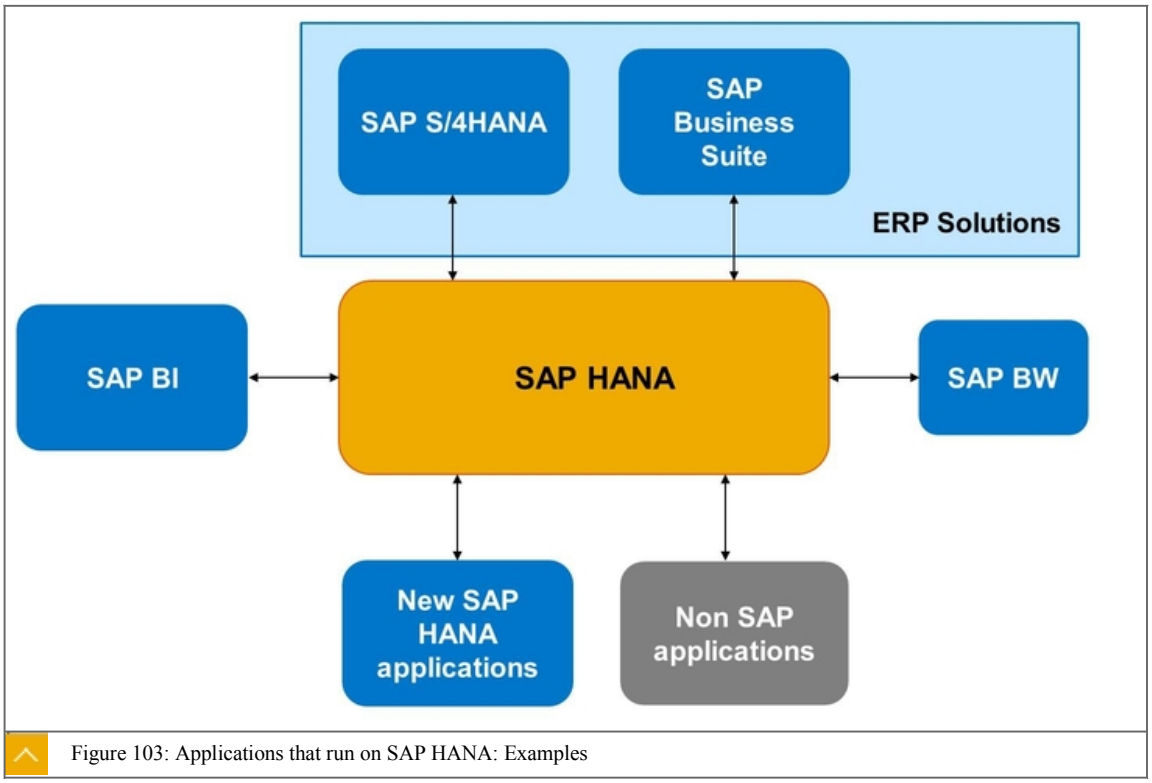


Figure 103: Applications that run on SAP HANA: Examples

SAP S/4HANA is SAP's totally rewritten Business Suite for SAP HANA. This flagship application covers all core business processes and applications for lines of business (LoBs). It is only available to run on SAP HANA. SAP S/4HANA includes a database client that communicates with SAP HANA, sending commands to the database for reads, inserts, delete, and updates. This is completely invisible to business users, who interact only with the applications from SAP S/4HANA that face them.

SAP Business Suite is the original business suite that was developed to run on a large number of enterprise databases. In recent years, SAP have made it possible for SAP Business Suite to run on SAP HANA. The underlying ABAP code was tweaked and optimized to ensure that everything still runs as expected and performance improvements can be realized. Business users do not see the interaction between SAP Business Suite and SAP HANA, and the database client of SAP Business Suite is the bridge between the application and SAP HANA. What business users experience is a massive speed-up of their applications.

SAP Business Warehouse (BW) is SAP's flagship enterprise data warehouse solution. As with SAP Business Suite, when BW was launched in 1998, it ran on well known disk based databases. But recently SAP BW can now run on SAP HANA and enjoys a massive boost in loading and reporting performance. SAP BW also integrates tightly with the core modeling of SAP HANA to provide sophisticated modeling options. SAP BW is written in ABAP, so the communication with SAP HANA is the same as with any other SAP applications, including SAP Business Suite.

SAP BI covers all the advanced analytical applications from SAP, which include the SAP BusinessObjects platform tools such as Web Intelligence, Crystal Reports, but also other SAP BI solutions including SAP Lumira and SAP Predictive Analytics. In almost all cases, the BI applications send requests to SAP HANA for data. They usually do not generate new data for SAP HANA. Depending on the analytic tool, there are different ways to connect to SAP HANA including OBDC, JDBC, etc.

New SAP HANA applications are a range of powerful, innovative, real-time applications that are for lines of business and are developed by SAP to run only on SAP HANA. Examples include, SAP Sales and Operation Planning, SAP Cash Forecasting, and SAP Collections Insight. These applications rely heavily on the inbuilt XS/XSA application server technology.

Non-SAP applications are developed by customers and partners using one or more platforms services from SAP HANA. Many of these applications are certified and promoted by SAP. There are many startups who have built mobile applications powered by SAP HANA. Remember that SAP HANA is available as a cloud service, and so even small startups can begin at low cost.

Connecting to SAP HANA

A key decision you must make when connecting any consumer to SAP HANA is whether the connection is made directly to the database or to the application server (XSA), which then, in turn, connects to the database. For custom applications, the developer needs to make this choice. For off-the-shelf applications, there is usually a predetermined method of connection that cannot be changed. For example, SAP BI applications such as Analysis for Microsoft Excel, connect directly to the SAP HANA database using a database connection technology (ODBC etc.). The SAP HANA application server (XSA) is not used.

When connecting directly to the database you must decide which database query language your application will use to communicate to SAP HANA. SAP HANA understands two database languages: **SQL** (Structured Query Language) and **MDX**(Multi Dimensional Expression). Both are powerful and popular industry-standard query languages.

SQL is a relational query language used by both transactional and analytical applications to not only read, but to write data back to a database. The strength of SQL is its universal popularity, which means there is a large global skill-set base.

MDX is also a query language, but it is used only by analytical applications to request data. It does not write data back to the database. MDX does have a key strength: It makes writing queries on dimensional (cube) data easy, as you can express a complex, multidimensional query with simple syntax, compared to SQL.

Off-the-shelf applications are usually fixed to use one of these query languages. For example, SAP S/4HANA uses SQL as its communication language between the application and the SAP HANA database. In contrast, Microsoft Excel Pivot Tables uses MDX as its communication language with SAP HANA.

Architecture for Connection to SAP HANA

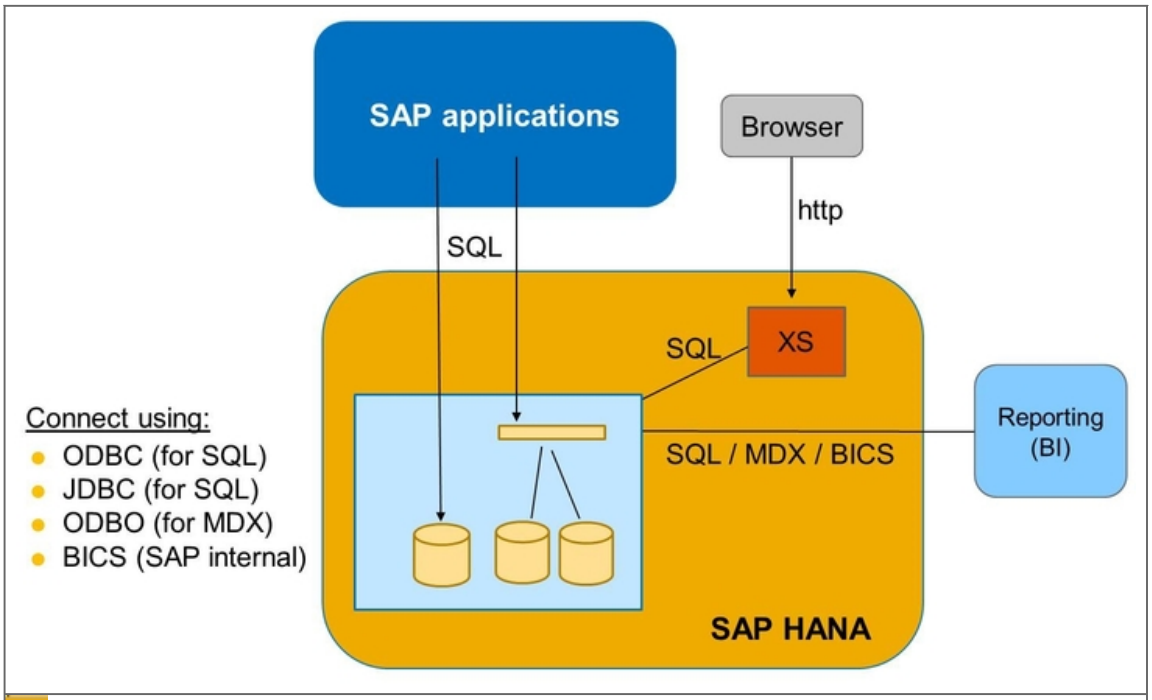


Figure 104: Architecture for Connection to SAP HANA

When connecting to the database, you should think about whether you wish to connect directly to the tables, or to the calculation views. Calculation views are usually recommended, especially when you want to push processing down to the database. Also, because calculation views contain reusable business logic, that means we don't have to reproduce this in the application layer.

Some applications do not allow consumption of tables directly (for example Analysis for Microsoft Excel), and so you must always create calculation views on top of the tables.

Another decision you must make is which connection technology to use to connect your application to the SAP HANA database. These are the most popular choices:

- ODBC

  This was developed by Microsoft for Windows-based applications that need connection to databases. ODBC uses SQL as the query language.

- JDBC

  This is a variant of ODBC, but it is only for JAVA-based applications. A key strength is that it is not tied to Windows applications, but can be used by applications that run on Unix and Linux. JDBC uses SQL as the query language.

- ODBO

  This is a Microsoft development specifically built for the MDX query language to connect OLAP clients to SAP HANA. Microsoft Excel Pivot Tables use this connection method.

- Business Intelligence Consumer Services (BICS)

  BICS was developed by SAP to provide optimal communications between its OLAP clients (such as Analysis) and SAP HANA. BICS offers the best performance and the most

comprehensive reach of OLAP functions, compared to ODBC, ODBO, and JBDC. However, only SAP can use BICS to connect its applications to SAP HANA as there is no public API for BICS for customer or partner use. It is a proprietary interface owned by SAP and its details are not published. It is mentioned in this section for completeness of knowledge.

Note:
BICS made its debut a few years ago when SAP wanted to improve the performance and functionality of the connection between SAP BusinessObjects reporting tools and SAP BW. Until then SAP was simply using industry standard technologies such as JDBC and ODBC.

While developers can choose the connection technology for their own applications, most off-the-shelf applications have a fixed method for connecting to SAP HANA database using one of these technologies. This cannot be changed. An exception to this would be SAP Crystal Reports, where ODBC and JDBC connections are offered.

LESSON SUMMARY
You should now be able to:

- Describe the types of application you can run on SAP HANA

Unit 5
Lesson 2

Running SAP Enterprise Suites on SAP HANA

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Run enterprise suites on SAP HANA

Running Business Suite on SAP HANA

The foundations of SAP Business Suite can be traced back to the early 1990s. Back then, the suite was called SAP R/3 and it consisted of a long list of application modules such as SD (sales and distribution), MM (materials management), and FI (finance) that supported key business processes. The application modules were powered by a technology platform called Basis, which provided all the underlying technology services needed to run the applications.

SAP R/3 was built to run on any of the leading enterprise relational database management systems (RDBMS) on the market. Back then, databases were disk-based, and tables were organized as row store. SAP R/3 was designed around the technology of that time, which meant many work-arounds were needed to maximize the performance of the system. For example, we needed to create separate aggregated tables to store summarized data, to help speed up management reporting. We also needed to build and maintain huge numbers of indexes to provide fast access to tables.

In the late 1990s, SAP R/3 was renamed ERP and was joined by many other key applications including CRM, SRM, and BW, to form a comprehensive suite of applications called SAP Business Suite. This significantly increased the range of business functions and processes available, and also provided solutions to connect customers and business partners in an emerging internet based world.

SAP Business Suite ran on a technology platform called SAP NetWeaver, the core of which is still the well-established Basis. However, SAP NetWeaver offered significantly more technology services than Basis, including data integration tools, and it also provided a modern web-services-based development infrastructure that supported service-oriented architecture (SOA).

The key principle of NetWeaver was to provide an enterprise-wide technology **platform** to not only power SAP Business Suite, but to support custom application development with all the tools needed in the SAP NetWeaver toolbox.

However, the underlying application design of SAP Business Suite and also NetWeaver services was still based around disk-based, row storage and also the hardware technology of that time. The work-arounds to help improve performance were still needed. In fact as time went on, the code and data model became increasingly complex. To help with performance, we moved data completely out of the applications and copied it each night to a dedicated storage component. This eased the pressure from the business transactional system and also enabled better performance for reporting users who were not competing for resources with the transactional users. However, the downside is that the IT landscape had become complex and data was being duplicated.

Business Suite Powered by SAP HANA Benefits

SAP HANA is a full relational database. It can be used wherever a relational database is needed to power any type of application. This includes SAP Business Suite. SAP wasted no time in making SAP HANA available as a database to run SAP Business Suite, so that customers could move away from the old disk-based, row store technology to a modern platform.

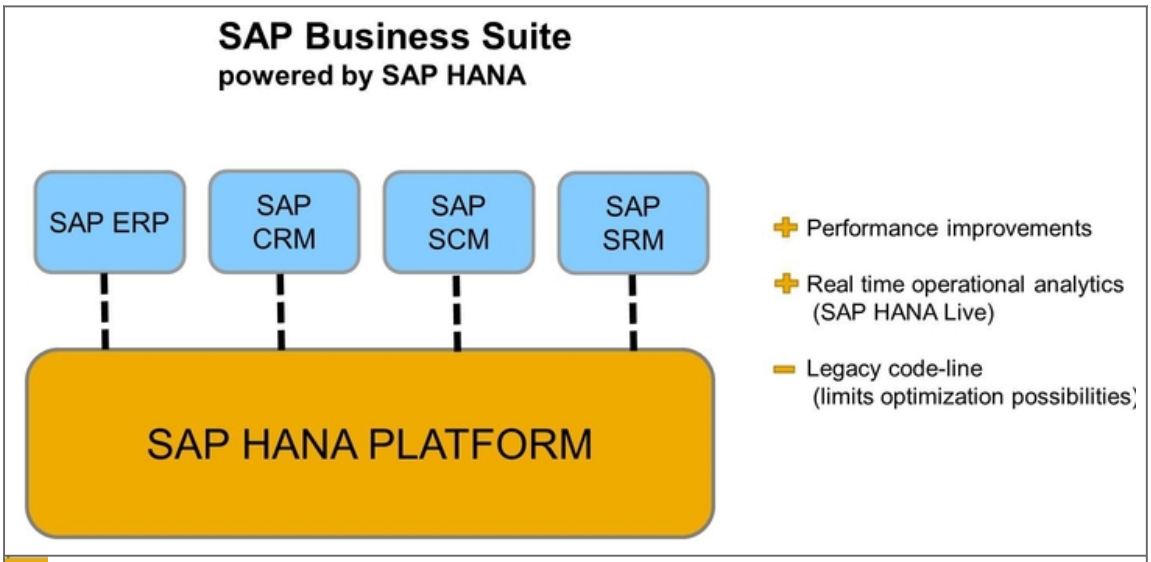


Figure 105: BS powered by HANA

SAP Business Suite powered by SAP HANA was born and provided immediate benefits. The following are just a few of those benefits:

- Massive speed-up of performance of existing applications
- A possibility to simplify the IT landscape by combining transactions and operational analytics back into one system
- Real-time reporting on operational data
- A new interface called SAP Fiori to modernize the user experience
- A platform that not only runs SAP applications, but provides a digital platform to power partner and customer developed applications

However, it must be remembered that the core code-line for SAP Business Suite was developed many years ago and its design was based on the technology at that time (disk based, limited memory, etc.). This means that we are not able to take advantage of potential optimizations and simplifications that come from SAP HANA. Also, bear in mind that SAP Business Suite code-line still has to be compatible with non-HANA databases, and that also limits what we can do with the code.

Today, large numbers of SAP customers have migrated the legacy databases that run their SAP Business Suite, so they now run on SAP HANA. SAP provides all the tools to make the migration easy and, mostly, automated.

Note:
Despite it being commonly used, SAP Business Suite on SAP HANA is not the official name. It is SAP Business Suite powered by SAP HANA.

SAP Business Suite still requires SAP NetWeaver to provide the underlying technology layer on the application side. SAP Business Suite is built using ABAP, and NetWeaver is needed to provide the ABAP development and runtime environment. However, with SAP Business Suite on SAP HANA, SAP have optimized the ABAP code to run on SAP HANA, to ensure that the best performance is possible. The data models remain the same, which means that the familiar ABAP table names are not changed, and custom developments continue to work with little or no adjustment.

To a business user, the migration is non-disruptive. There are new SAP Fiori interfaces available, but these are optional and users can continue to work with SAPGUI as before. The biggest change for the user is the massive improvement in performance of applications, especially batch processes such as MRP and period end preparation and closing. They also benefit from access to real-time reports and dashboards on operational data.

SAP S/4HANA Runs on SAP HANA

SAP S/4HANA (Suite for HANA) is SAP's next-generation business suite for the digital world. It is built to run only on SAP HANA. It is not available on any other third party database or platform such as Oracle, IBM, or Microsoft.

Many customers have migrated from Business Suite on anyDB to Business Suite on SAP HANA to obtain a massive speed-up in performance of the application. The move to SAP S/4HANA represents the next phase in their transition to a next-generation business platform.

For customers who did not migrate from Business Suite on anyDB to Business Suite on SAP HANA, they can upgrade and migrate straight to SAP S/4HANA. Customers running non-SAP enterprise suites can implement SAP S/4HANA and take advantage of the supplied tools to support the data migration.

Code for SAP S/4HANA

SAP S/4HANA was completely rewritten. We still use ABAP for the application code, but it has been rewritten to exploit the full power of the SAP HANA in-memory processing engines and database. By contrast, Business Suite on SAP HANA uses the same ABAP code as Business Suite on anyDB, but it was adjusted to ensure that it could run on SAP HANA.

A rewrite for SAP S/4HANA was needed because the code line for Business Suite was written a long time ago (and the coding approach fit the technology of that time). There were many approaches used to ensure best performance that are simply not needed today with SAP HANA. For example, SAP S/4HANA does not use stored aggregates to summarize data. SAP HANA can summarize data on the fly. As well as this, SAP S/4HANA does not need indexes, as the design of the column store database means that we already have fast access to any combination of data columns.

Business Suite had to support multiple databases, so the code was complex to ensure compatibility.

SAP S/4HANA Powered by SAP HANA

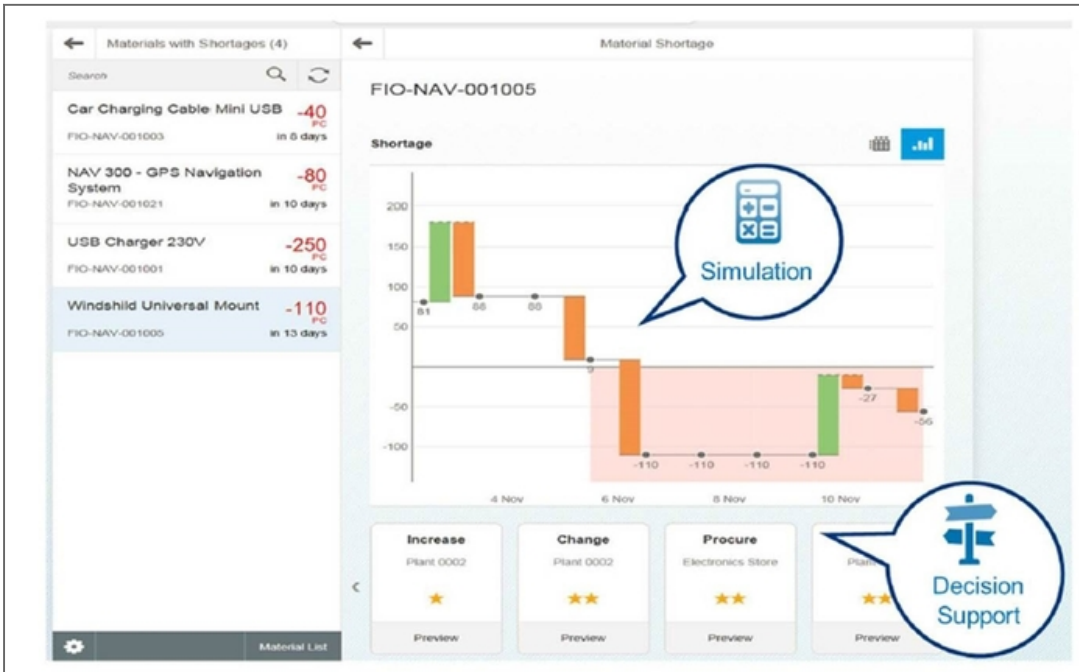


Figure 106: SAP S/4HANA Powered by SAP HANA

Like Business Suite on SAP HANA, SAP S/4HANA provides a new interface called SAP Fiori, which improves performance and user experience. SAP Fiori applications are intuitive and simple to use, with a modern look and feel. SAP Fiori applications run on any device. Unlike SAPGUI, users are no longer tied to their desktops, so they can switch devices and work with the same applications.

Some SAP S/4HANA Components

A key component of SAP S/4HANA is Embedded Analytics. Embedded Analytics provides a complete data model, based on Core Data Services (CDS). It also provides tools covering all application areas that can be used to build real-time operational analytics, with no additional tools needed.

SAP provide thousands of consumption-ready views of real-time business data. SAP S/4HANA includes an in-built analytic engine to handle complex OLAP requests in memory. It also includes easy-to-use query buildings and reporting interfaces.

A major difference between Business Suite and SAP S/4HANA is the provision of many brand new applications. Business Suite on SAP HANA focuses on continuity. SAP S/4HANA provides continuity, while also providing next-generation digital applications that combine analytics and transactions. SAP have rebuilt many applications so that they provide the business user with analytics, right in place where decisions are needed.

LESSON SUMMARY
You should now be able to:

- Run enterprise suites on SAP HANA

Unit 5
Lesson 3

Connecting SAP Business Intelligence Tools to SAP HANA

LESSON OVERVIEW
This lesson covers connecting Business Intelligence (BI) tools to SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Connect SAP Business Intelligence Tools to SAP HANA

SAP Business Intelligence (BI)

Before we begin to examine the details of connectivity to SAP HANA, let's provide some basic background on SAP BI.

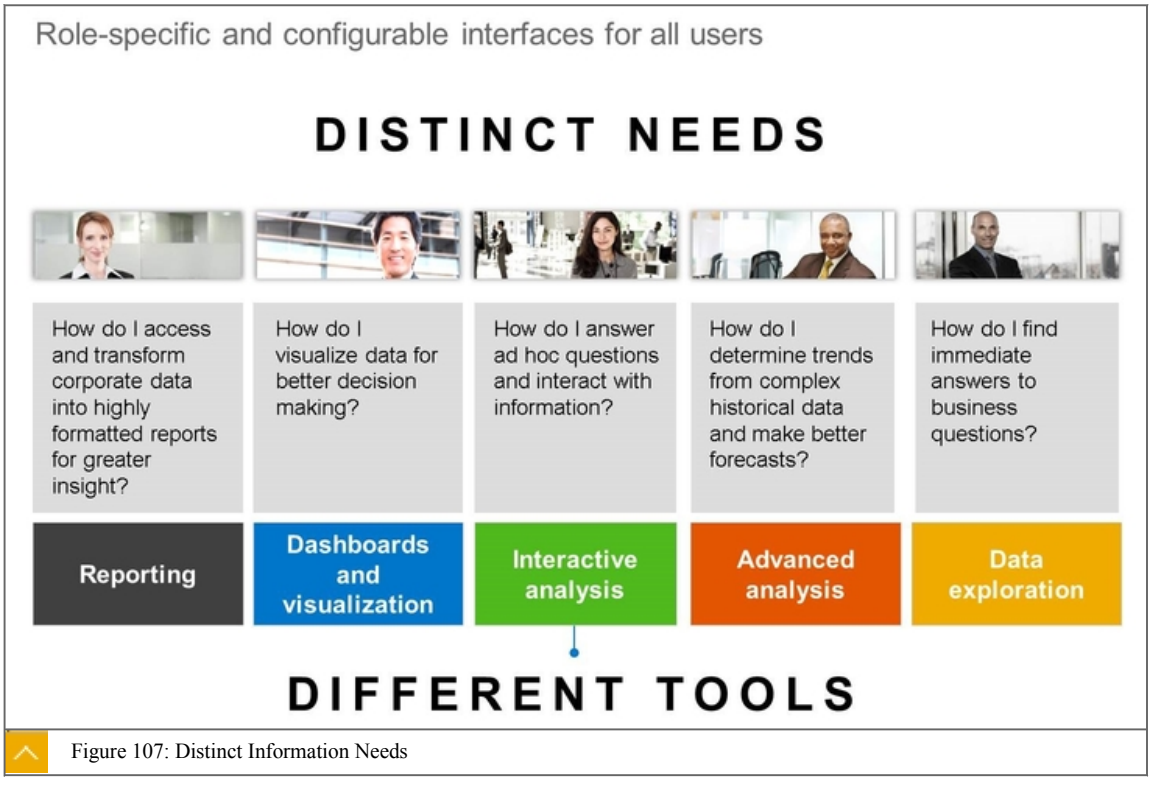


Figure 107: Distinct Information Needs

In any business organization, different roles have varying needs for information. Some users expect a weekly, well formatted, and consistent report, and some require access to a dashboard where they can get up-to-date information on Key Performance Indicator (KPIs). Some users require the ability to drill down and explore their data with no boundaries, and some users need to be able to prepare and develop their own on-demand analysis. SAP provides a dedicated tool for each type of user.

Information Needs Matched To Clients

Many of the BI tools have their origin in the SAP BusinessObjects family and have been enhanced and natively integrated with SAP solutions (SAP BW, SAP HANA). Some BI tools were developed from scratch by SAP, such as SAP BusinessObjects Lumira and Analysis, Edition for Office.

SAP BI is the name of the portfolio of these tools.

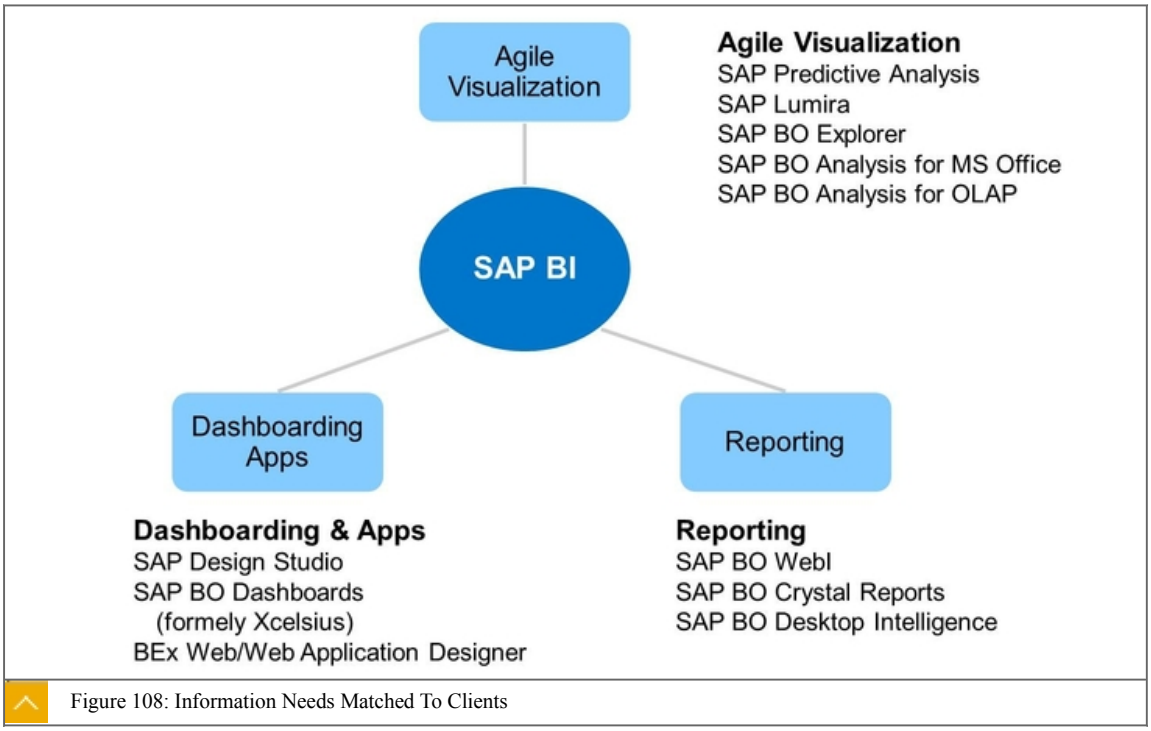


Figure 108: Information Needs Matched To Clients

The Convergence of the SAP BI Client Portfolio

There are many tools in the SAP BI family so let’s group them by type.

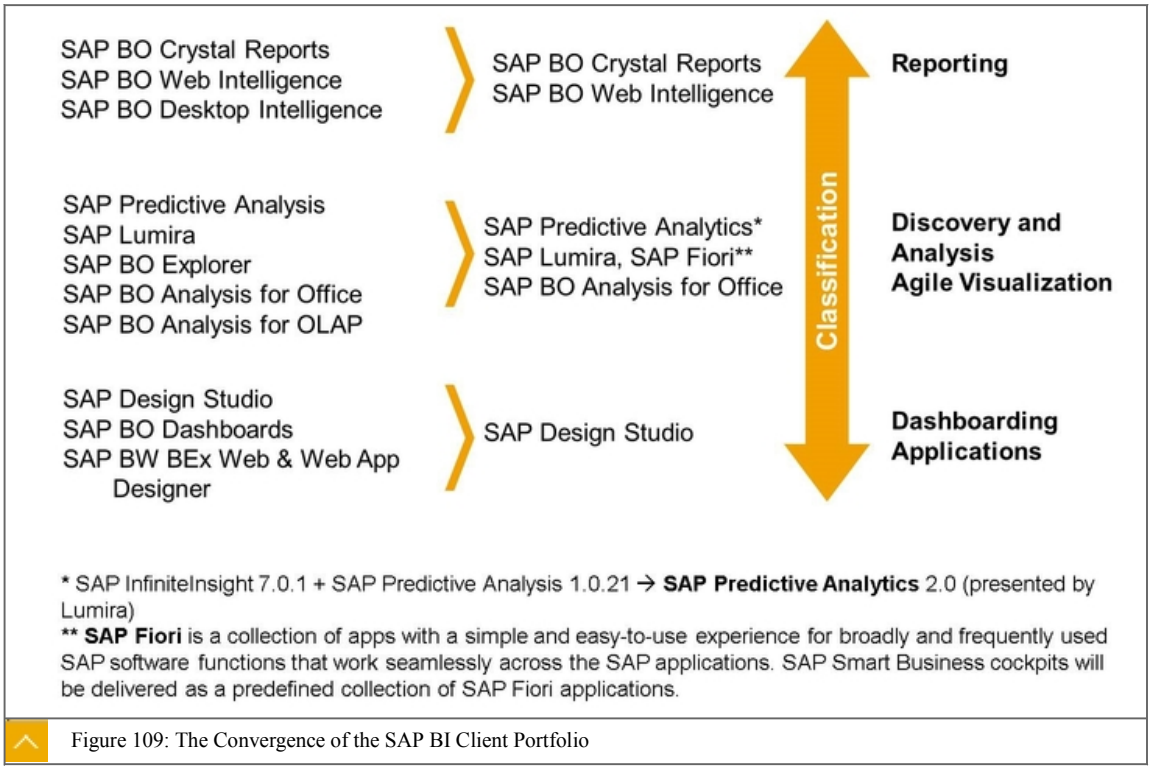


Figure 109: The Convergence of the SAP BI Client Portfolio

Recently, SAP have simplified the landscape for SAP BI by consolidating the functionality of tools. We now have fewer tools, but the tools that are left are more powerful.

Let's take a quick look at each SAP BI tool and pick out some strengths of the tool as well as some key points relating to SAP HANA.

Connecting SAP BI to SAP HANA

One of the earliest use cases for SAP HANA was to power real-time operational analytics. Many customers have deployed SAP HANA as a data mart, to capture data from various operational systems, including SAP enterprise systems. SAP HANA is ready to connect to a huge number of reporting tools, including SAP BusinessObjects, using industry standard connectors. So, reporting on SAP HANA can begin immediately.

One of the most appealing aspects of SAP HANA in this use case is that SAP HANA comes with extensive built-in virtual data models. These models provide real-time business-ready views of all SAP operational data, based on the tables of either SAP Business Suite or SAP S/4HANA. SAP have developed and maintained these comprehensive virtual data models to expose live operational tables from all areas of SAP Business Suite and SAP S/4HANA. This means that any application that requires easy-to-consume views of business data has it immediately.

The real value in the virtual data models is the business semantics added by SAP. Raw database tables are combined, and filters and calculations added, to expose business views ready for immediate consumption with no additional modeling needed. So, instead of having to refer to multiple raw tables in your reporting tool, creating joins and unions manually and applying filters to add meaning to the data, you can simply call a view from the virtual data model and the data is exposed.

SAP-Supplied Virtual Data Models

For SAP Business Suite, the virtual data model is called SAP HANA Live and it is specifically built for SAP Business Suite tables. For SAP S/4HANA, the virtual data model is delivered

using Core Data Services (CDS). CDS is delivered with SAP S/4HANA as part of Embedded Analytics and it is based specifically on the simplified table schema of SAP S/4HANA.

While they are different technical approaches, they both deliver the same outcome; that is, a virtual data model that exposes live operational data for analytics.

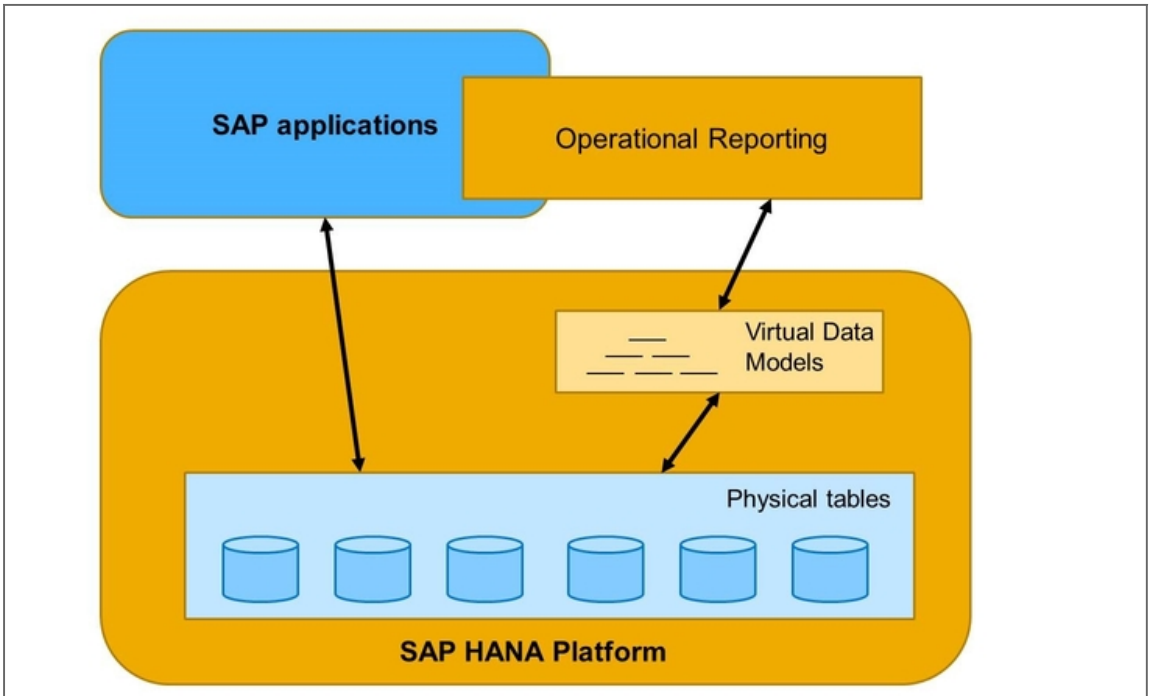


Figure 110: SAP-Supplied Virtual Data Models

Get Started with SAP HANA Live

Customers who run SAP Business Suite on SAP HANA or AnyDB can get started immediately with SAP HANA Live. Customers do not need to convert their SAP Business Suite to run on SAP HANA to use SAP HANA Live. Customers can continue to use their existing SAP Business Suite applications on AnyDB, and SAP HANA Live can be installed on a side-by-side SAP HANA platform.

This means that data must then be replicated from the source tables from AnyDB into SAP HANA, preferably in real time. This can be achieved in a variety of ways using standard SAP data replication tools.

For customers already running SAP Business Suite on SAP HANA, no data replication is needed. This is because SAP HANA Live is installed on the same SAP HANA platform used to power SAP Business Suite. This means SAP HANA Live has access to the same tables used by the transactional applications.

To get started with the virtual data model built with Core Data Services, customers must implement SAP S/4HANA, as the virtual data model is supplied only with SAP S/4HANA Embedded Analytics.

Connecting Tools

Almost all reporting clients and tools are able to connect to SAP HANA directly. The reporting tool that cannot directly connect to SAP HANA is SAP BusinessObjects Dashboards. For this tool, you first need to create a Universe. Each reporting tool connects to SAP HANA using one of the following:

- ODBC — generic relational
- JDBC — JAVA relational
- ODBO — multidimensional

  =

- BICS — SAP proprietary for multidimensional

Some tools offer a choice of connectors. For example, with Crystal you can use either a JDBC or ODBC connector.

The query language used depends on the reporting tool used and the connection type. The query languages are SQL (for relational models) and MDX (for multidimensional models).

SAP BusinessObjects Analysis Edition for Microsoft Excel

SAP BusinessObjects Analysis Edition is a popular reporting tool for powerful OLAP exploration, with no boundaries. It is perfect for exploring the multidimensional SAP HANA models that we created in an earlier unit, with lots of slice and dice.

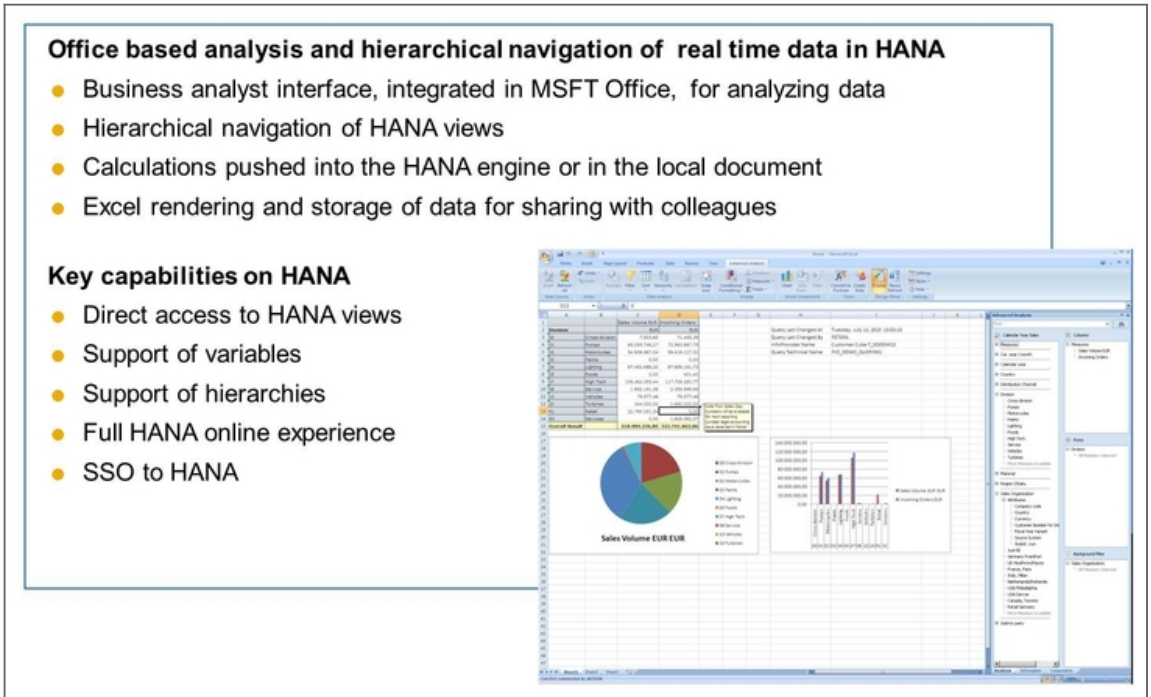


Figure 111: Advanced Analysis: SAP BusinessObjects Analysis for MS Office

Connecting to SAP HANA

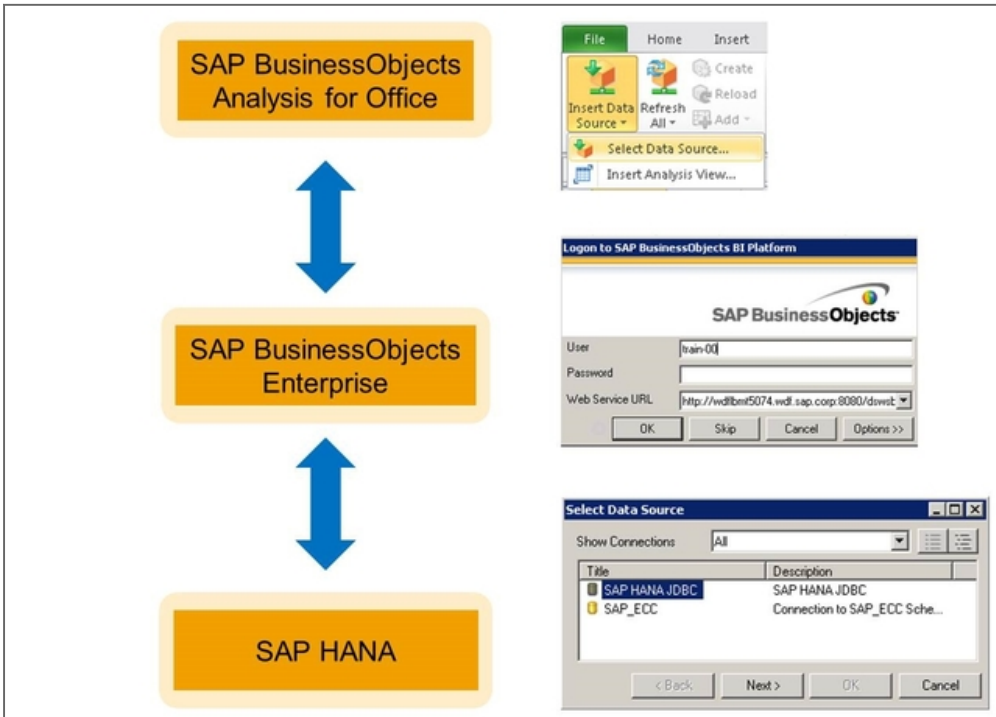


Figure 112: Connecting to SAP HANA

The methods for connecting Analysis Office to SAP HANA are as follows:

- You can use a secured, shared connection that the administrator has previously defined in the SAP BusinessObjects Enterprise platform.

  When opening Analysis, you enter (or confirm) a web service URL that points to the SAP BusinessObjects Enterprise platform connection. You also enter your SAP BusinessObjects Enterprise platform user and password.

  The key point of this method is that you do not have to know any SAP HANA users and passwords to reach SAP HANA data sources. It also means that existing SAP BusinessObjects users can use their existing user and password.

- You can use a local connection that is not hosted by SAP BusinessObjects Enterprise platform.

  This method requires the user (or admin) to manually add a connection from Excel (on the local connection dialog) that points directly to SAP HANA. To set this up you need the host name and instance of the SAP HANA system you want to connect to.

  When the user clicks on this connection, they are prompted for a valid SAP HANA user and password.

The Design Panel

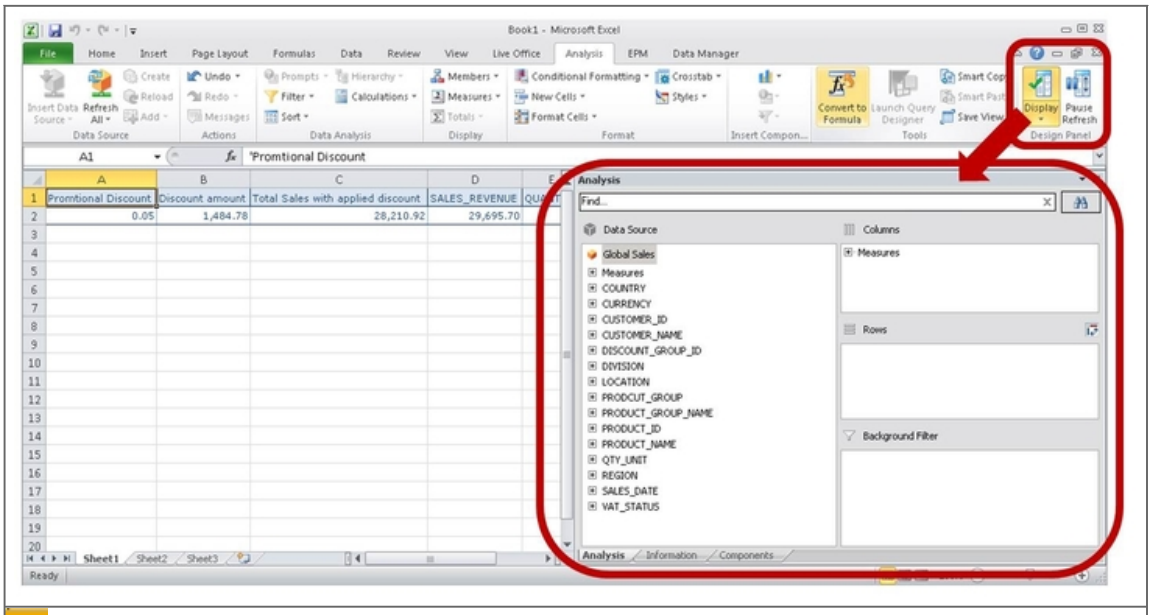


Figure 113: The Design Panel

The default view when inserting an SAP HANA data source, such as an analytic or calculation view, is a display of all available measures with no attributes.

The data is displayed in the workbook in cross tabs. You can insert multiple cross tabs in a workbook with data from different sources and systems. If the workbook is to be used by different users, it is helpful to add info fields with information on the data source and filter status.

Using the design panel, you can analyze the data and change the view of the displayed data. You can add and remove dimensions and measures to be displayed easily with drag and drop.

To avoid single refreshes and data fetch from SAP HANA after each step, you can pause the refresh to build a cross tab. After ending the pause, all changes are applied at once.

Unit 5
Exercise 9

Consume a Calculation View with SAP Analysis for Microsoft Excel

Exercise Objectives

In this exercise, learn how to build a report that consumes a Calculation view of type CUBE.

Business Example

You have been tasked with creating a flexible report in Analysis for Microsoft Excel on top of a HANA Calculation view. The report is to be used to analyze sales to US customers, with the possibility to drill-down to examine sales by any combination of customer and product attributes.

Note:
In this exercise, when values include ##, replace these characters with your own student number.

1. Start Analysis for Microsoft Excel and open a blank workbook.
2. Insert a new data source based on the ready-made SAP HANA calculation view US Product Sales which is found in the folder HA100_00_HDI_CONTAINER_1 → HA100 . Use the logon details from the table, Logon Details . Make sure cell A1 is selected before you begin.

   Table 13: Logon Details

Field	Value
Connection	HTTP_HANA
User	**STUDENT##**
Password	**Training1**

3. Display the Quantity and Amount by Delivery attribute.
4. Break down the data by Product Group .
5. Remove the Delivery charge type and now break down the Product Group by customer Location in the columns to create a cross tab report.
6. Close Analysis for Microsoft Excel .

Unit 5
Solution 9

Consume a Calculation View with SAP Analysis for Microsoft Excel

Exercise Objectives

In this exercise, learn how to build a report that consumes a Calculation view of type CUBE.

Business Example

You have been tasked with creating a flexible report in Analysis for Microsoft Excel on top of a HANA Calculation view. The report is to be used to analyze sales to US customers, with the possibility to drill-down to examine sales by any combination of customer and product attributes.

Note:
In this exercise, when values include ##, replace these characters with your own student number.

1. Start Analysis for Microsoft Excel and open a blank workbook.
   a) In the Windows Start page, begin typing **`Analysis`** and in the search results on the right of the screen, choose Analysis for Microsoft Excel .
   b) At the next prompt, double-click Blank workbook .
2. Insert a new data source based on the ready-made SAP HANA calculation view US Product Sales which is found in the folder HA100_00_HDI_CONTAINER_1 → HA100 . Use the logon details from the table, Logon Details . Make sure cell A1 is selected before you begin.

   Table 13: Logon Details

Field	Value
Connection	HTTP_HANA
User	**`STUDENT##`**
Password	**`Training1`**

   a) Make sure that cell A1 is selected.
   b) From the Excel toolbar, choose the Analysis tab.
   c) On the Analysis ribbon, choose Insert Data Source .
   d) From the menu, choose Select Data Source .

e) In the next dialog, choose Skip to avoid logging on using the SAP BusinessObjects Platform connections.

f) In the next dialog, choose the connection from the table, Logon Details , and choose Next .

g) In the next dialog, enter the user and password provided in the table, Logon Details .

h) In the Select Data Source dialog box, choose the Area tab.

i) Expand the folder HA100_##_HDI_CONTAINER_1 → HA100 and choose your calculation view US Product Sales .

j) Choose OK.

3. Display the Quantity and Amount by Delivery attribute.

a) Drag the Delivery charge type attribute from the Data Source pane to the rows pane below.
The measures are now split by the two types of Delivery Charge: chargeable and free-of-charge .

4. Break down the data by Product Group .

a) Drag the PRODUCT_GROUP_TEXT attribute from the Data Source pane to the rows pane below.
The measures are now split by Delivery charge type and Product Group.

5. Remove the Delivery charge type and now break down the Product Group by customer Location in the columns to create a cross tab report.

a) Drag Delivery charge type from the rows to the Data Source pane to remove it from the results.

b) Drag Location from the Data Source pane to the columns pane, above.

6. Close Analysis for Microsoft Excel .

a) Close Analysis for Microsoft Excel .

SAP Crystal Reports

SAP Crystal Reports is a well established tool, used to develop sophisticated business reports. Its strengths include the following:

- Pixel-precise formatting
- Any data sources, local or enterprise
- Dynamic content
- Multiple output formats, such as HTML, PDF, XL, or Crystal Viewer

For example, when sales figures are low, the reports can automatically include additional breakdown figures to explain the situation. Reports are of a professional standard, so they can be presented for external consumption (for example, to present shareholder information on the company's website).

SAP Crystal Reports is a desktop tool and it is regarded as the world's most popular business reporting tool. SAP Crystal Reports can connect to any data source, SAP and non-SAP. It is possible to use SAP Crystal Reports to connect to SAP HANA, using ODBC or JDBC industry standard connectors.

SAP Crystal Reports for Enterprise

A few years ago, SAP developed a special version of SAP Crystal Reports called SAP Crystal Reports for Enterprise. This version can natively connect to SAP enterprise data sources, such as SAP BW and SAP BusinessObjects Universes.

SAP Crystal Reports for Enterprise does not connect to SAP sources using industry standard connectors such as ODBC. Instead it uses SAP's own connection technologies, such as Business Information Consumer Services (BICS). The key reason we use BICS is so that SAP can consume all the SAP features and content of the data sources, to provide the best performance. Standard connectors (ODBC, JDBC, and so on) often do not support every feature of SAP sources, which is why SAP decided to create their own connection technology.

SAP Crystal Reports and SAP Crystal Reports for Enterprise

SAP Crystal Reports and SAP Crystal Reports for Enterprise can connect directly to SAP HANA or they can connect indirectly via a Universe. SAP Crystal Reports can access the SAP HANA database tables directly or the Calculation views. SAP Crystal Reports for Enterprise can only access Calculation views, not tables directly.

Both versions of SAP Crystal Reports support many of the modeling features of SAP HANA Calculation views, such as variables and hierarchies.

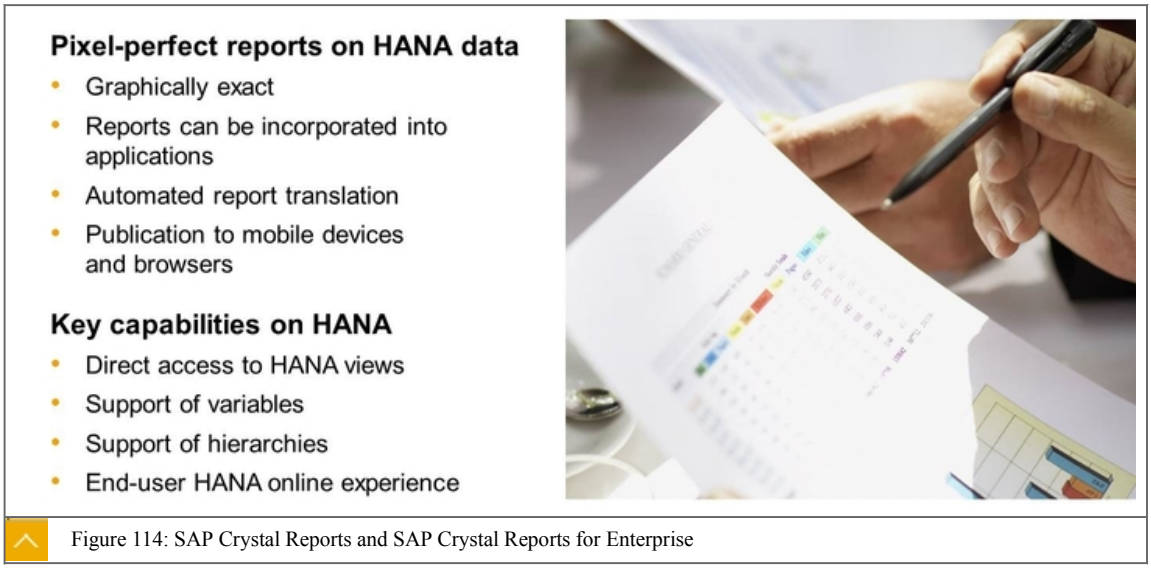


Figure 114: SAP Crystal Reports and SAP Crystal Reports for Enterprise

SAP BusinessObjects Web Intelligence

SAP Web Intelligence is a powerful report building tool. It is aimed at business users who would like to use a self-service BI tool.

SAP Web Intelligence, usually referred to as Webi, is ideal for creating on-demand reports. Its simple user interface has been welcomed by many business users who enjoy being able to create their own reports without the need for specialist BI skills.

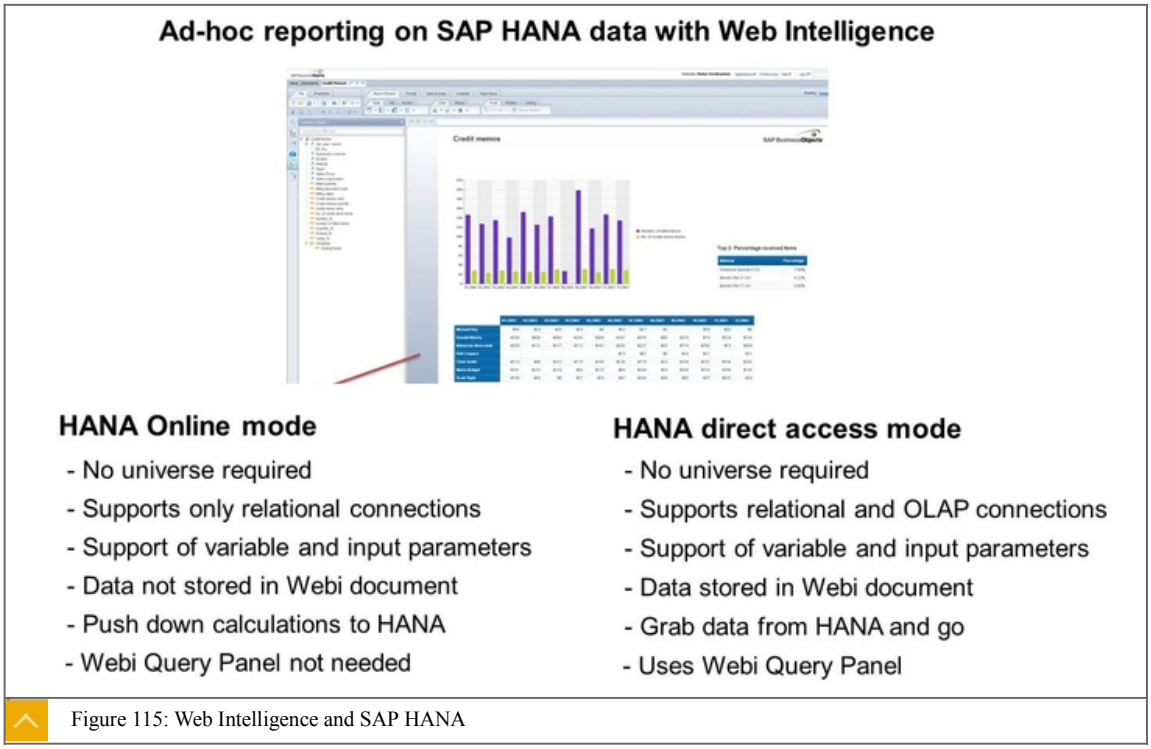


Figure 115: Web Intelligence and SAP HANA

Reports are created using a simple drag and drop approach, and the interface, which is aimed at a business user, is easy to use. Webi is often positioned as an on-demand reporting tool, used to build queries for one time use. However, many of our SAP customers tell us that Webi is the most important reporting tool in their organization due to its power and ease of use.

Until recently, Webi was one of the few tools that could not directly connect to SAP HANA. Therefore, a universe was always needed to act as the communication layer between Webi

and SAP HANA. However, since SAP BusinessObjects 4.2 and SAP HANA SPS10, Webi can connect directly to SAP HANA, and a universe is not needed.

The modes required when connecting to SAP HANA are as follows:

- Online mode

  Webi is always connected to SAP HANA to ensure great performance by pushing down processing to HANA.

- Direct Access mode

  Webi grabs data from SAP HANA and can then go offline. Then, processing is done in Webi.

Note:
With regard to SAP HANA hierarchies, it should be noted that because online mode only uses a relational connection, SAP HANA hierarchies will be flattened and treated as dimensions. With direct access mode, which can use OLAP connections, the SAP HANA hierarchies are structurally preserved and appear in the Webi Query Panel as hierarchies.

SAP BusinessObjects Design Studio

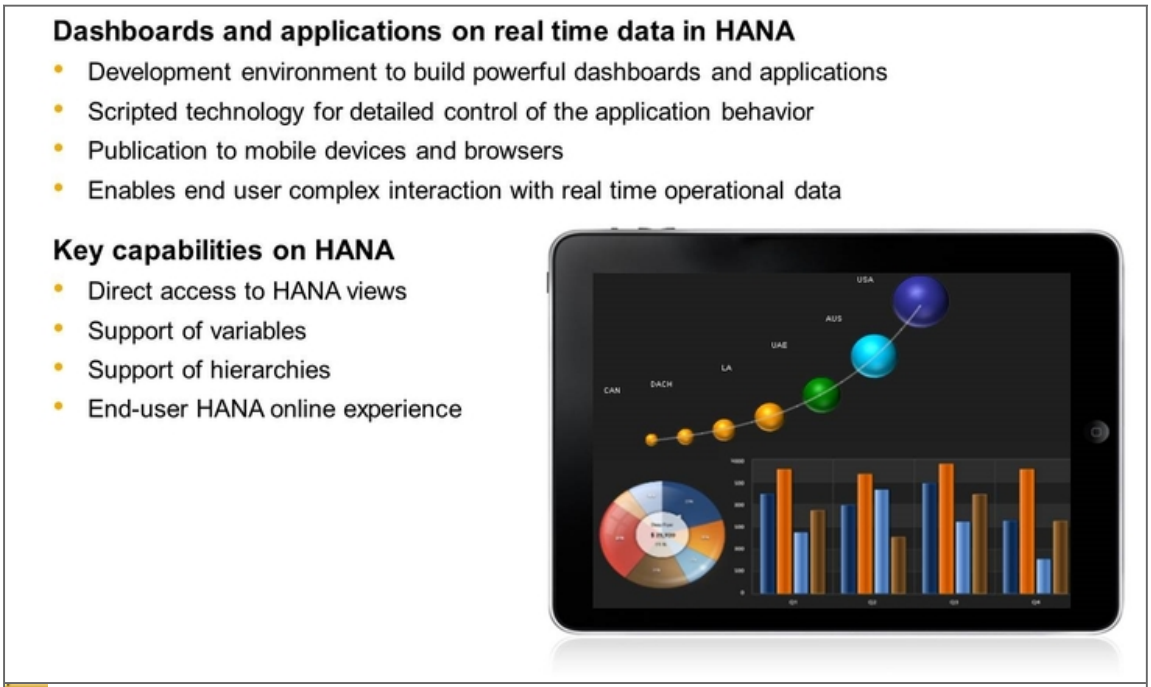


Figure 116: SAP BusinessObjects Design Studio

SAP Design Studio is the flagship dashboard and cockpit development tool that is used to develop powerful mashups for desktop and mobile devices. SAP HANA can provide data directly to the dashboards for real-time insights.

Note:
SAP Design Studio has recently been merged with the latest version of SAP Lumira (version 2.0) and is now a ‘mode’ of Lumira called **SAP Lumira Designer** .

SAP Lumira

SAP BusinessObjects Lumira delivers beautiful analytics, allowing you to easily convey and share knowledge in the following ways:

- Use beautiful and intuitive data visualization templates.
- Share visualizations and insights in the cloud with everyone, without having to sacrifice performance, security, or functionality.
- Combine and mashup all your data with simpler, error-free data prepping workflows. Sophisticated charts and visualizations illustrate hidden insights where tabular reports are not effective.

Data Search and Exploration: SAP BusinessObjects Lumira

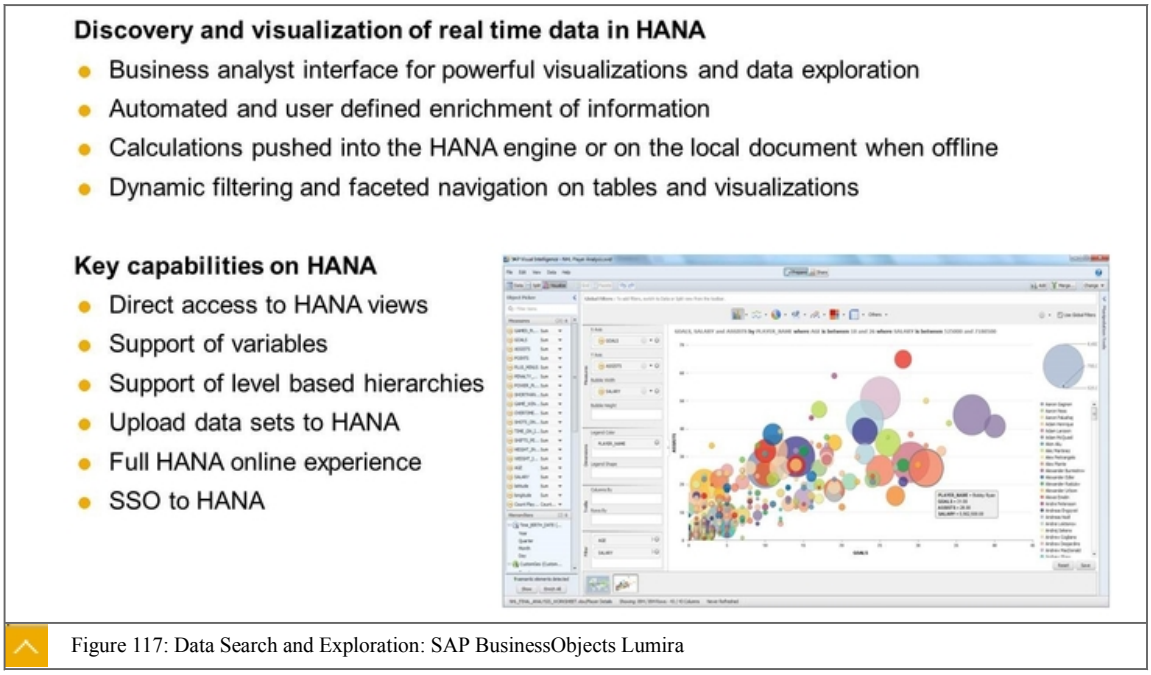


Figure 117: Data Search and Exploration: SAP BusinessObjects Lumira

SAP BusinessObjects Lumira is a self-service solution that allows analysts and decision makers to access, transform, and visualize data.

The SAP BusinessObjects Lumira desktop experience is used to prepare data from multiple sources and visualize it. You can then compose stories from those visualizations, which can be shared with other decision makers using the SAP BusinessObjects Lumira server and cloud platforms that provide browser and mobile-based experiences. From here you can further analyze data and collaborate with colleagues on datasets, stories, and other business intelligence artifacts.

SAP Lumira 2.0 represents a significant milestone in the Lumira evolution. SAP Lumira 2.0 is a merge of the Lumira 1.x functionality and also SAP Design Studio.

There are two modes for SAP Lumira 2.0:

- **Discovery** — this brings in the original SAP Lumira functionality
- **Designer** — this brings in the SAP-Design-Studio-like functionality.

Both modes allow connectivity to SAP HANA, either for acquiring the SAP HANA data as a one off task, or by having a live data connection.

LESSON SUMMARY
You should now be able to:

- Connect SAP Business Intelligence Tools to SAP HANA

Unit 5
Lesson 4

Outlining SAP Business Warehouse powered by SAP HANA

LESSON OVERVIEW
This lesson covers SAP Business Warehouse (BW) on SAP HANA.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Outline how SAP BW leverages SAP HANA

SAP Business Warehouse (BW) powered by SAP HANA

SAP Business Warehouse (BW) was the first SAP application enabled to run on SAP HANA. In recent years, large numbers of customers have been migrating their disk-based legacy databases under BW, to SAP HANA. This immediately provides them with superior performance from their SAP BW solutions, particularly in the key areas of reporting and data loading. It also opens up powerful data modeling possibilities by combining BW and core HANA modeling techniques.

Note:
SAP HANA was developed using a lot of the technology first developed for Business Warehouse Accelerator (BWA), the in-memory appliance that improves read performance of InfoCubes.

SAP BW Simplified when Running on SAP HANA

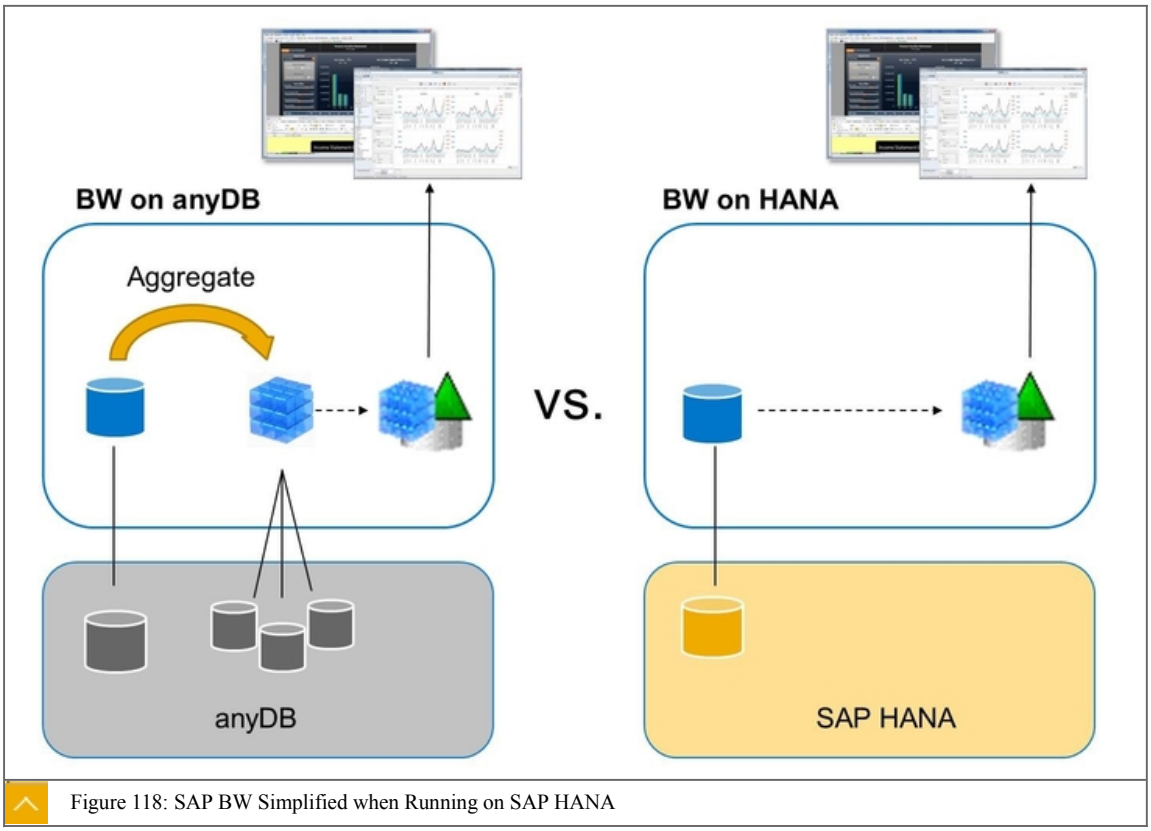


Figure 118: SAP BW Simplified when Running on SAP HANA

With SAP BW powered by SAP HANA, it is not just the speed that improves dramatically. Other major benefits include a significantly reduced data footprint.

A non-HANA SAP BW design involves building many layers of stored aggregations. We do this in order to improve the speed of reporting by precalculating business data in advance of the users requesting it. With the raw power of SAP HANA, aggregations are done on the fly when the users runs a query, so you no longer need to build and store aggregates.

By removing the aggregates, which are usually modeled using BW InfoCubes, you not only reduce the data footprint, but also dramatically simplify the data modeling tasks. The data loading is quicker when you do not need additional layers. Plus, it is easier to implement new flows and integrate them into existing models when you only need to manage one layer of data and not have to consider layers of aggregations.

With BW powered by SAP HANA, the only persistent layer is modeled with Advanced DataStore Objects (ADSO). These are only available with BW powered by SAP HANA and are optimized for in-memory processing.

Integrate any Data with BW

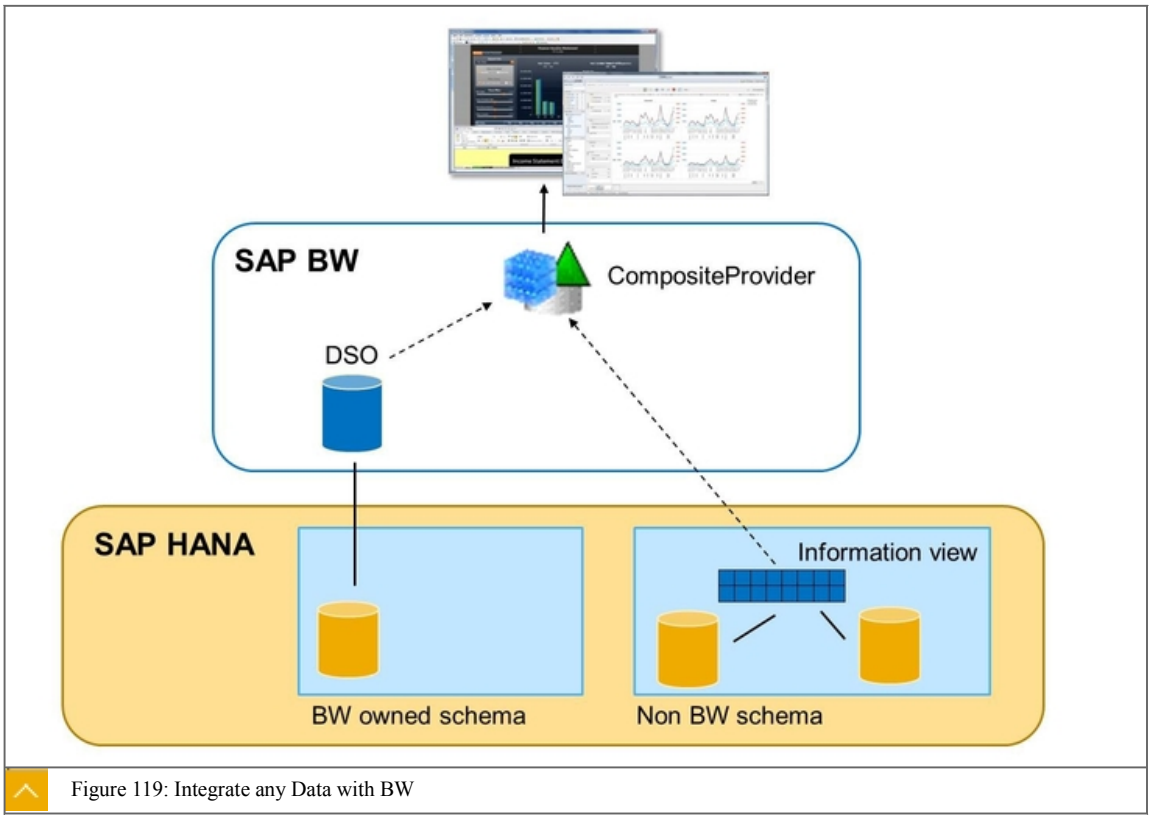


Figure 119: Integrate any Data with BW

When running SAP BW powered by SAP HANA, you have access to all external SAP HANA schemas. You can leverage the advanced data acquisition mechanisms that are part of SAP HANA, such as SDA, SDI, SLT, and streaming, to provision data to these external schemas. You can then build SAP HANA Calculation views over the tables in the external schemas and integrate these views with SAP BW modeled objects, such as CompositeProviders.

This means that you can combine the data warehouse data, perhaps representing the historical data, with SAP HANA data in external schemas representing live operational data. SAP BW reports and applications can then consume the combined data, giving you a complete picture of historical and up-to-date data for your real-time applications. This combination of the modeling features of BW and SAP HANA is called mixed modeling.

Integrate BW Data with any Application

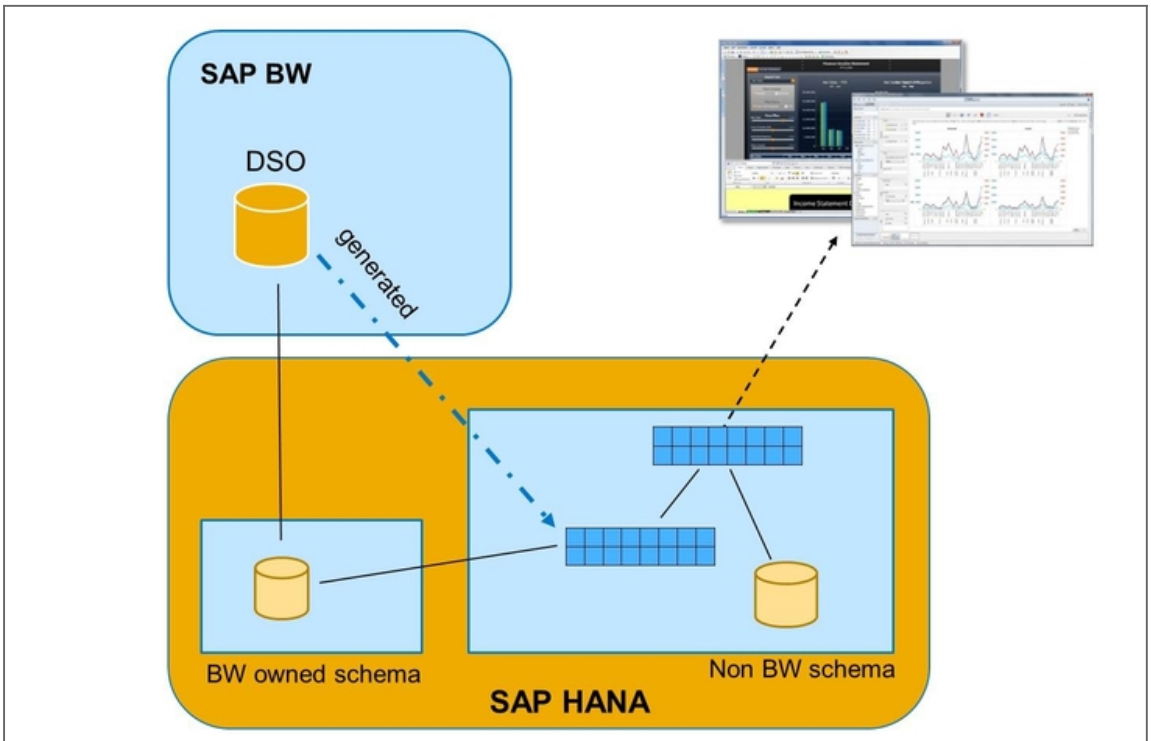


Figure 120: Integrate BW Data with any Application

SAP BW is able to automatically generate SAP HANA calculation views that correspond to InfoCubes, DataStore Objects, and other BW modelled objects. You can then integrate these generated SAP HANA calculation views into other SAP HANA views, which could provide additional calculations, to create a mash-up that can be consumed by any tools (including re-consuming by SAP BW).

This means that you can expose SAP BW data to any application that can connect to SAP HANA. If you change the design of SAP BW objects, the corresponding SAP HANA calculation views are automatically regenerated so everything is kept in step.

BW Powered by SAP HANA Versus SAP BW/4HANA

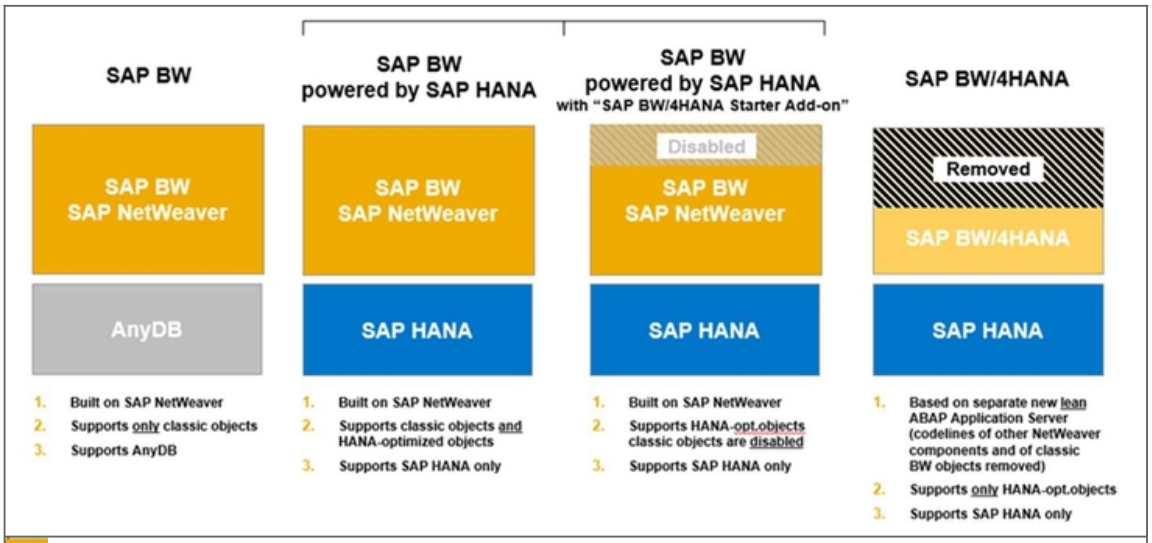


Figure 121: BW Powered by SAP HANA Versus SAP BW/4HANA

There are two versions of BW that can run on SAP HANA. They are as follows:

- **SAP BW powered by SAP HANA**

  This is supported for BW 7.4 and BW 7.5, and is popular with customers who are transitioning from BW on anyDB. With this edition, it is possible to create and maintain the legacy BW objects, such as InfoCubes, alongside the new HANA optimized objects, such as Advanced DSO and CompositeProviders.

  Modeling, query building, and administration is carried out using the traditional SAPGUI interfaces. Eclipse can also be used for some of this. Both 3.x and 7.x data flows are supported in this version.

  A few years after introducing BW powered by SAP HANA, SAP developed an add-in that can be installed. Its purpose is to shut down the classic objects of BW. Customers do not have to install the add-in, but if they do, it eliminates the risk of BW objects being created that are not optimized for HANA. This is often done as part of the preparation for the move to BW/4HANA, where classic objects are not supported. The last version of SAP BW powered by SAP HANA was 7.5.

- **SAP BW/4HANA**

  This is a brand new BW written from scratch. It represents SAP's next-generation data warehouse. Although it is built on the principles of classic BW, this edition supports only SAP HANA-optimized objects. Therefore, huge amounts of ABAP code, and features that were not needed, can be removed, to create much leaner BW.

  With this version it is only possible to create and maintain SAP HANA-optimized BW modeling objects, such as CompositeProviders. InfoCubes, MultiProviders are no longer supported.

  BEx Query Designer is not used. Queries are built using the Eclipse tools and SAPGUI is not needed for modeling. Customers can upgrade from BW powered by SAP HANA to BW/4HANA using supplied tools. However, customers must first remove or migrate all classic objects to the new SAP HANA-optimized objects. SAP BW/4HANA is a new product and so has a new version numbering that started at 1.0.

You can compare BW powered by SAP HANA to Business Suite powered by SAP HANA, and BW/4HANA to S/4HANA. The former are code adjustments and maintain support for non-SAP HANA databases. The latter are complete code rewrites and support only SAP HANA.

Note:
SAP BW powered by SAP HANA is often referred to as SAP BW on HANA, but this is not the official title. Also BW/4HANA is still new to the market and is often confused with BW powered by SAP HANA. Make sure you don't confuse these.

LESSON SUMMARY
You should now be able to:

- Outline how SAP BW leverages SAP HANA

Unit 5
Lesson 5

Describing SAP HANA Data Warehousing Foundation

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe SAP HANA Data Warehouse Foundation

Native Data Warehousing with SAP HANA

For many people, when they hear references to SAP data warehousing, they immediately think of SAP BW powered by SAP HANA, and more recently, BW/4HANA. Both of these remain SAP's recommended solution for enterprise-wide data warehousing. However, SAP HANA is now mature enough and also contains the basic components required to build a custom data warehouse from scratch, without the need for SAP BW.

Let's describe the different approaches:

SAP BW is an application-driven data warehouse solution. This simply means that all tooling and services, extractors, data governance, and modeling objects needed to support a data warehouse are already built by SAP and are ready to go. SAP BW is tightly integrated in to SAP enterprise suites and other SAP applications and that makes it very appealing to existing SAP customers.

A data warehouse built using only SAP HANA is called an SQL-driven data warehouse. Here, you use SAP HANA tooling to build all the services from scratch. A comprehensive data warehouse can be built but it requires a lot of effort. But the benefit is that is can be completely build to a customer specification. This approach might also be appealing to customer who do not already have SAP applications.

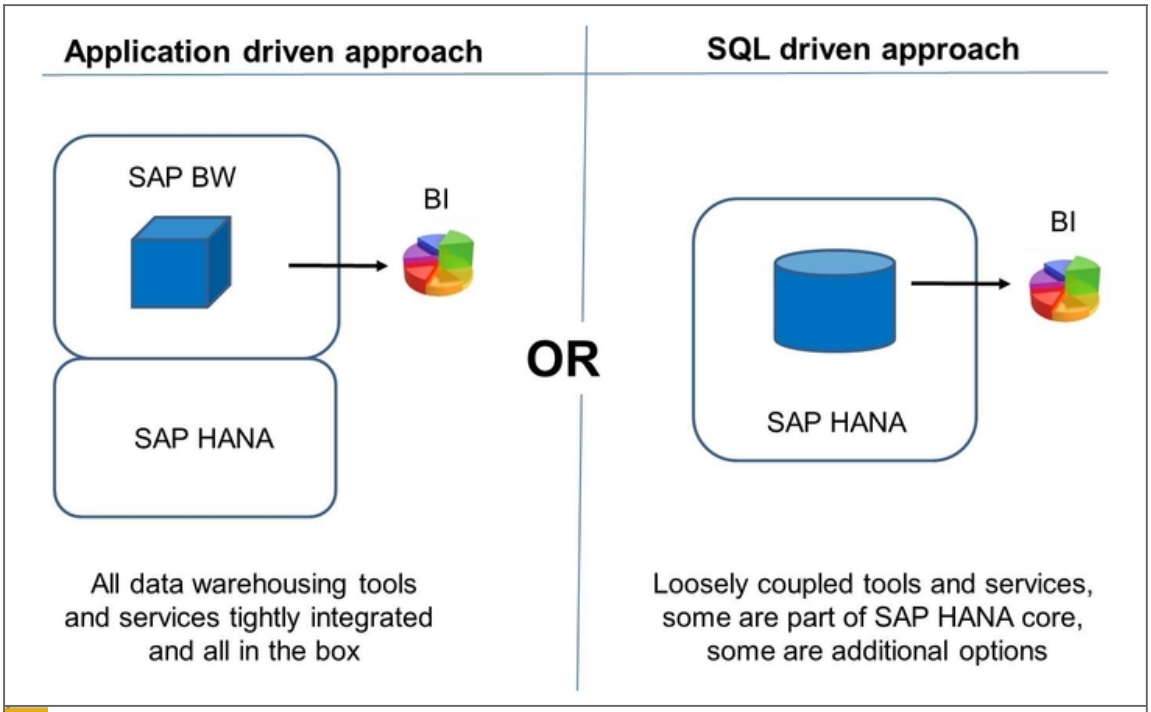


Figure 122: Choosing a Data Warehousing Approach

SAP HANA Components

Since the introduction of SAP HANA in 2011, SAP has always provided OLAP modeling and query processing capabilities in the SAP HANA platform in order to support BI cases. Over the years, SAP has been introducing more complex OLAP capabilities that first appeared in BW to SAP HANA. These functions include time-dependent hierarchies, dynamic parameter derivations using routines, etc.

So, for the front end, SAP HANA OLAP is now strong. However, other key components to manage and automate data distribution (LSA) are less well developed compared to BW.

However, this is starting to change and already SAP have delivered a brand new data storage component called a **Native DataStore Object** which provides data load request management. This is vital to be able to distribute data load-by-load into the data warehouse and also to handle delta loading using change logs that generate delta images of data. It is almost identical to the basic functionality of an SAP BW DataStore Object.

In addition, SAP have developed some native HANA tooling that can be installed to support data distribution and data lifecycle management.

Just so we are clear, let's explain what we mean by this:

- Data Distribution

  Ensuring that tables and partitions that should be processed together are always physically stored together on the same nodes in a scale-out cluster. This is to ensure best performance.

- Data Lifecycle Management

  Manage data temperatures by relocating data across storage tiers.

SAP have developed components that are installed into SAP HANA, which provide key data warehousing functions. These components are based on SAP HANA XS.

SAP HANA Data Warehousing Foundation: The Tools

SAP HANA Data Warehousing Foundation is a family of tools that support intelligent data distribution and management for large scale, SAP HANA-based data warehouses.

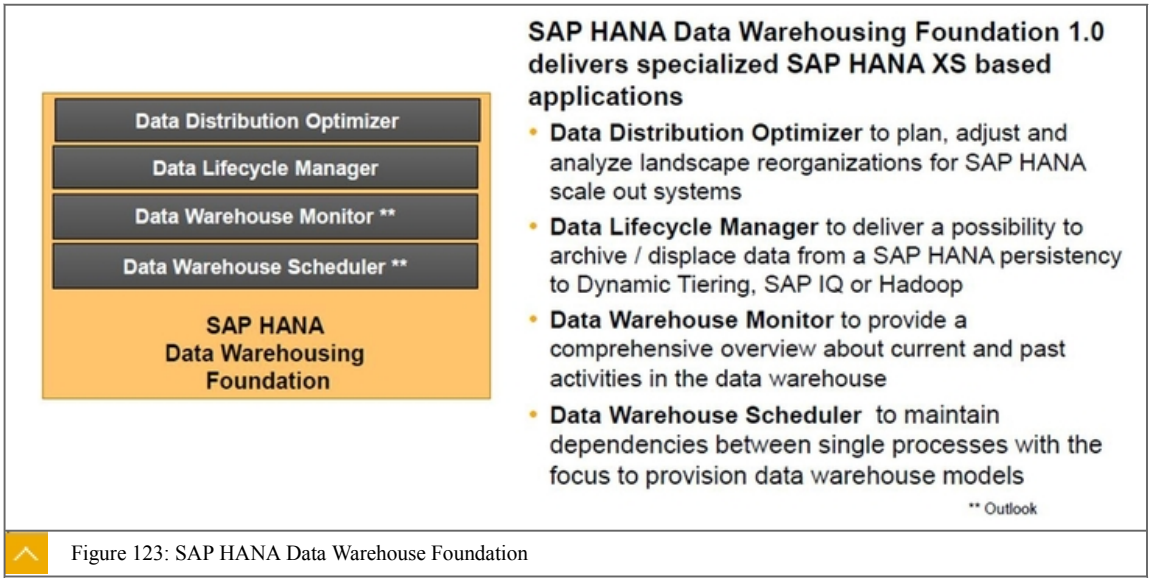


Figure 123: SAP HANA Data Warehouse Foundation

SAP HANA Data Lifecycle Management (DLM)

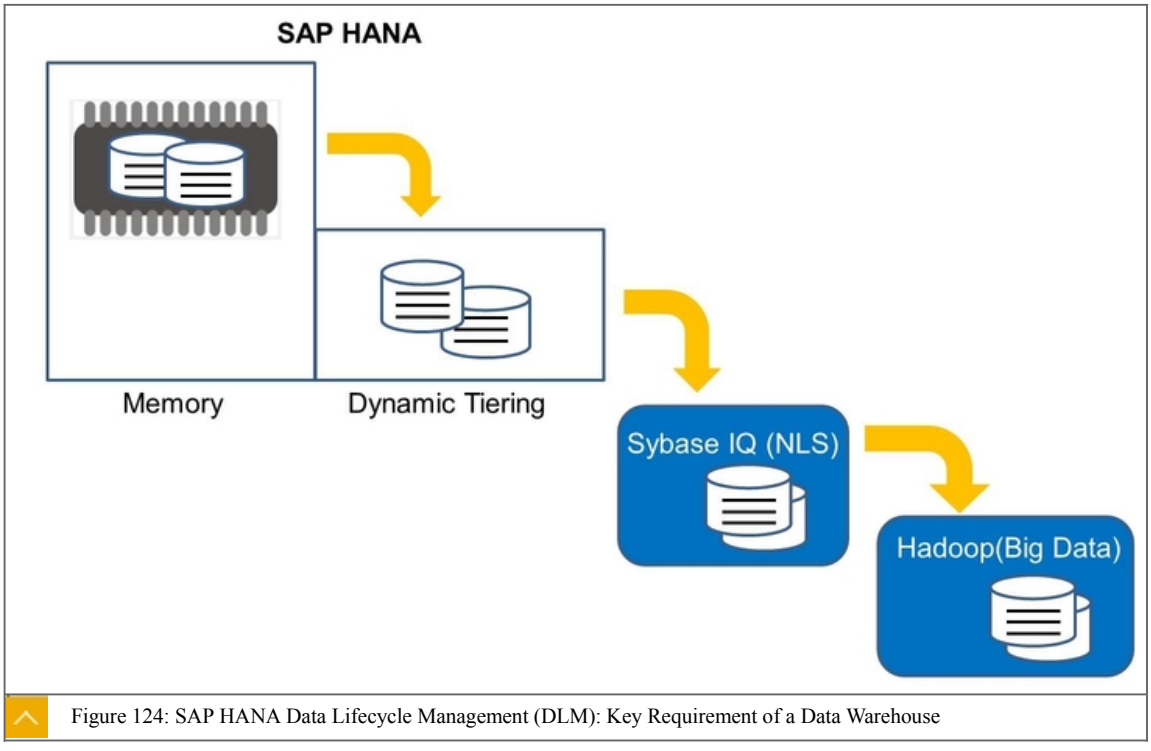


Figure 124: SAP HANA Data Lifecycle Management (DLM): Key Requirement of a Data Warehouse

With SAP HANA Data Lifecycle Management (DLM), you can explore your data across the entire landscape. You can do this by using graphical drill-down tools to identify tables that are appropriate for relocation. You can also use tools to forecast data growth, to ensure that you do not run out of memory or disk.

Data can be distributed to the following destinations:

- SAP HANA Dynamic Tiering
- SAP IQ over SDA
- Spark SQL for Hadoop
- Deletion Bin

You create predefined relocation rules. These rules define the conditions under which data should be moved and to where the data should be relocated.

SAP HANA Data Distribution Optimizer (DDO)

On a large scale, SAP HANA landscape data is spread out and stored across various nodes. This can mean that tables and partitions that are usually processed together, such as orders and returns, can find themselves located on separate nodes. The problem here is that joins must be carried out across nodes, which is not optimal for performance.

Ideally, tables that are processed together should be on the same nodes. The SAP HANA DDO tool provides analysis of the use of join paths. This makes it easy to identify how tables have been processed together so they can be assigned to the same nodes for improved performance.

SAP will release additional tools in the next releases to manage job schedules and monitor data warehousing activity.

Note:
DWF is not included in the core SAP HANA platform and is an additional installed option. Also, DWF has its own development cycle and version numbering. As of early 2017, the latest release is DWF 1.0 SPS05.

LESSON SUMMARY
You should now be able to:

- Describe SAP HANA Data Warehouse Foundation

Unit 5
Lesson 6

Building Native HANA Applications

LESSON OVERVIEW
In this lesson we will cover some of the basic concepts behind the creation of native SAP HANA applications using the XS engine.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe the basics of native HANA applications

Building Native SAP HANA Applications

When we refer to SAP HANA applications we need to differentiate between applications that are **powered by S/4HANA** and **native** HANA applications. An application that is powered by 4HANA usually has its own application server. What these applications need from HANA is the powerful data processing services. This is the case for SAP S/4HANA, Business Suite powered by HANA and SAP BW powered by HANA. These applications are all written in ABAP and so they require an ABAP application server. NetWeaver provides the ABAP application services. SAP HANA is not an ABAP application server.

Native Applications on SAP HANA

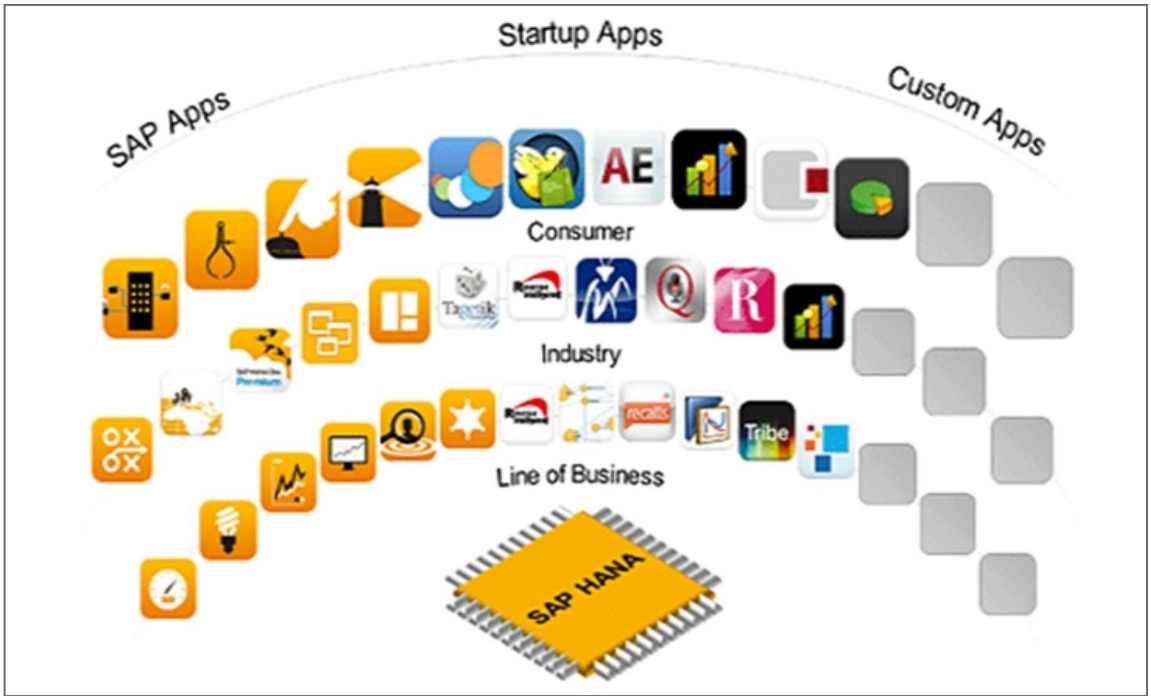


Figure 125: Native Applications on SAP HANA

But just like NetWeaver, HANA can provide sophisticated application services too. With HANA, you have all the tools we need to build brand new innovative applications from scratch. You can also run them and scale them across the enterprise.

In fact, this has been the case for many years and there are many SAP HANA applications that have been built by SAP, SAP partners, independent start-ups, and of course, customers.

Building Applications for SAP HANA

With HANA, you can build applications that run on-premise or in the Cloud, or both. Developers can use SAP HANA Cloud Platform (HCP) to develop and run their applications on a public platform. Alternatively, developers can use an on-premise version of HANA to develop and run their applications, which gives them control over the landscape.

SAP have built an active application development community who share ideas and code, enabling them to learn from each other.

When you build applications on HANA, you not only use the database and data processing engines, but also the application engine of HANA and all in a single tier. And because all of these run in memory, performance is improved versus disk-based applications that need to cross tiers.

Code Push-Down to SAP HANA

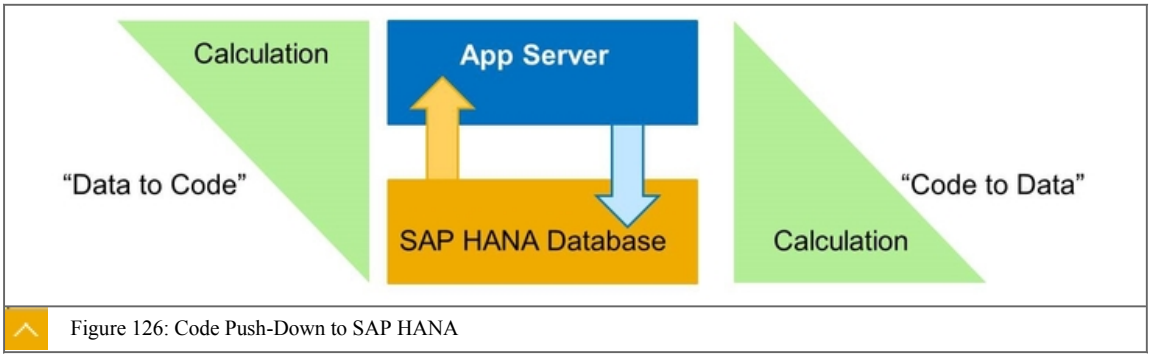


Figure 126: Code Push-Down to SAP HANA

SAP HANA native applications offer a natural way to push down the code. This is possible as the application and all data processing tasks run in memory, to enable excellent performance.

SAP HANA Application Development and Deployment

When approaching native SAP HANA application development, you need to recognize that there are two different development and deployment infrastructures with SAP HANA. They are as follows:

- XS

  Extended Application Services (also known as XS classic or XSC). Introduced with SAP HANA 1.0 SPS05 and still available in SAP HANA 2.0 for backward compatibility.

- XSA

  Extended Application Services Advanced. Introduced in SAP HANA 1.0 SPS11 and the future direction. XS is still supported but will not be developed any further.

Regardless of the physical infrastructure approach, the basic conceptual architecture of a native SAP HANA application remains the same.

Two-Tier Architecture of SAP HANA Application Development

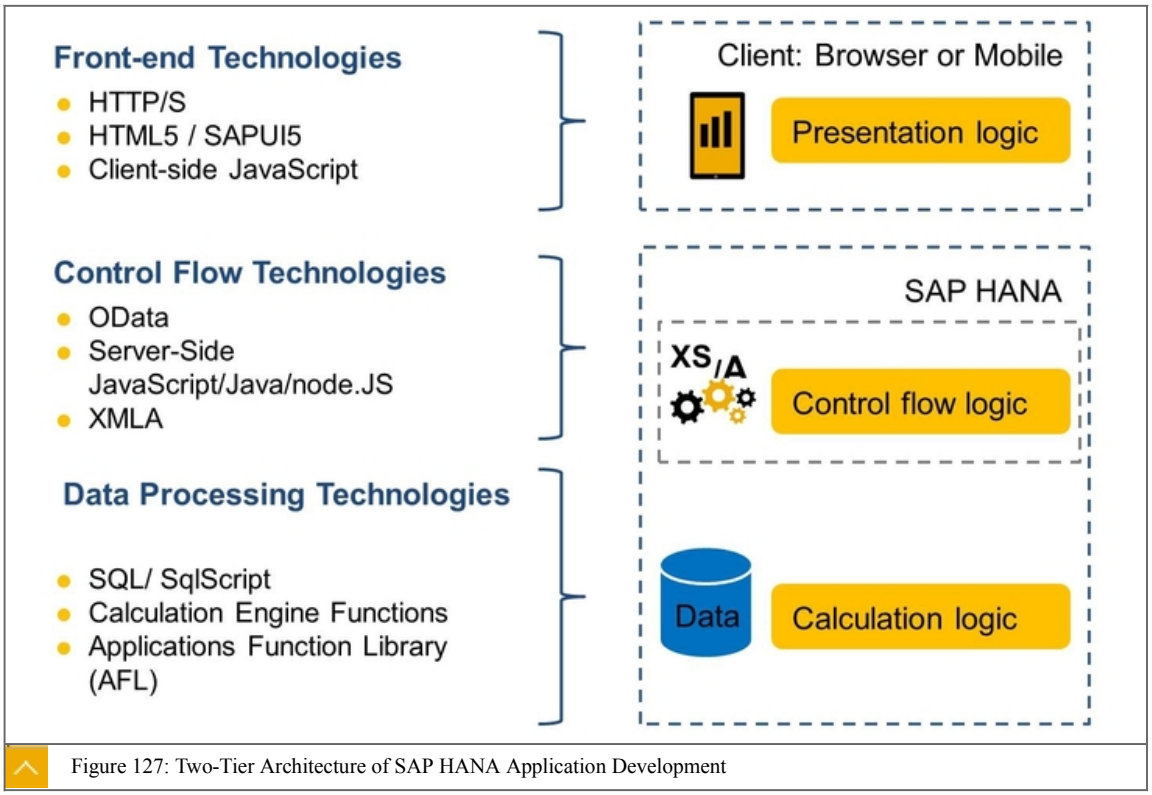


Figure 127: Two-Tier Architecture of SAP HANA Application Development

SAP HANA provides all the tools needed on the development side such as code editors, debuggers, source code version control etc. SAP HANA also provides the deployment and runtime for the various languages.

When you build native SAP HANA applications, you can simplify the stack by building the application logic and the database logic inside SAP HANA. You do not need an additional application server. All you need is a client that connects directly to SAP HANA XS/XSA using HTTP.

Building Applications on XS

In this section we focus on XS, the original application development and runtime for SAP HANA. It has been replaced with XSA but we should still develop our knowledge around XS, as many applications are built using XS and will need to be supported. Also, XSA is built on many of the principles of XS, so it is a good place to start.

XS is a small-footprint application server that sits completely inside SAP HANA. XS is not a separate technology that happens to be installed on the same hardware server as SAP HANA. It is actually an extension of, and tightly integrated into, the SAP HANA Database.

SAP have developed several applications that are built on XS (for example, SAP Operational Process Intelligence, SAP Smart Meter Analytics). With the release of SAP HANA SPS05, XS became available for customers and partners who wish to develop their own SAP-HANA-based applications.

Rationale for XS

The rationale for XS is as follows:

- Simplicity

Enable SAP-HANA-based application development and deployment, while minimizing architectural layers

- Performance

  Tight integration with the SAP HANA DB ensures optimal performance

With XS, you can create applications that have an HTTP-based end-user UI (for example, browser, mobile apps). These applications run directly on SAP HANA without additional external servers or system landscape components.

XS Architecture

This simplified system architecture means low TCO, as complexity of administering or dealing with other heterogeneous components is eliminated. Other advantages include efficient interprocess communications taking place inside one overall system, and same data types. These provide performance advantages that no other application server can match when SAP HANA is the DB.

XS even includes its own internal repository for content lifecycle management. This can be used for all kinds of development artifacts and objects.

These applications can utilize data within SAP HANA views and the applications can also read and write to tables directly.

XS Programming Model

The programming model, when working directly with XS, dictates that applications are designed in the following manner: Front-end processing should be delegated to the browser (or mobile device) using HTML5 and Client-Side JavaScript.

The vast majority of application logic, business logic, calculations, or any data-intensive operations should be implemented using SQL, SqlScript, or Calculation Engine (CE) functions. This way, the heavy lifting of application processing is performed in the DB, close to the data, and where SAP HANA is optimized for performance.

This leaves a relatively modest role for the application server and web server part of the architecture (XS). This role is mainly comprised of handling control flow logic, the “glue” that is necessary to tie together the user interface with backend DB processing. The technologies available for this purpose are Server-Side JavaScript, OData (Open Data Protocol), or XMLA (eXtensible Markup Language for Analysis).

XS in SAP HANA

Using the Developer perspective of the SAP HANA Studio, all of the aforementioned technologies can be utilized to build custom SAP-HANA-based applications of all kinds.

The XS Engine provides applications and application developers with access to the SAP HANA database, using a consumption model that is exposed with HTTP. In addition to providing application-specific consumption models, SAP HANA XS also hosts system services that are part of the SAP HANA database. Examples of these include search services and a built-in Web server that provides access to static content stored in the SAP HANA repository.

You can use OData to enable clients to consume authorized data stored in the SAP HANA database. OData defines operations on resources using RESTful HTTP commands (for example, GET, PUT, POST, and DELETE) and specifies the URL syntax for identifying the resources. Data is transferred over HTTP using either the Atom (XML) or the JSON (JavaScript) format.

XS Architecture Components

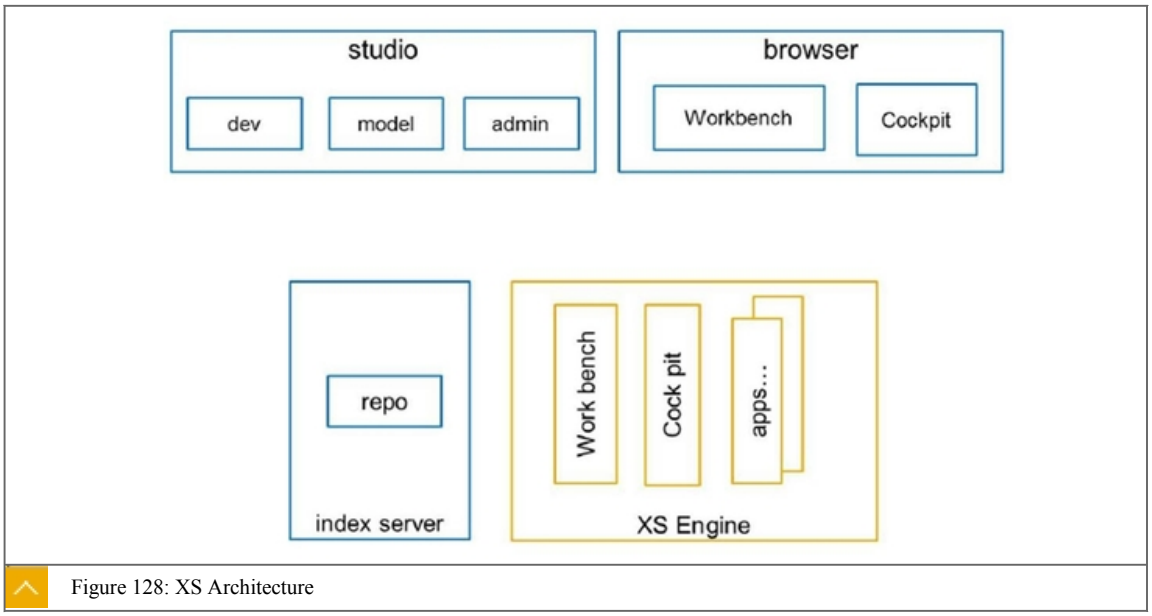


Figure 128: XS Architecture

In XS Classic, the following components are used to develop native SAP HANA applications:

- SAP HANA Studio, for administration and development
- SAP HANA web-based workbench, for application development
- Deployment, via Delivery Units
- SAP HANA Cockpit, for administration
- Design time repository inside SAP HANA
- Web server and XS Engine, for supporting web content and JavaScript runtime execution

Building Applications on XS Advanced

Since SAP HANA 1.0 SPS11, SAP introduced a new version of XS call XS Advanced (XSA). XSA is a huge advance on XS and introduces a number of major improvements for development and deployment of native HANA applications. These improvements allow customers to build and maintain applications much faster and also in more development languages than with XS. It is now possible to develop a single HANA application that can be deployed to the cloud and also on-premise. It is also possible to deploy multiple versions of the same application. This means upgrades to an application can be released whilst continuing to run older versions.

XSA uses a new interface called SAP WebIDE for SAP HANA for all development. Studio and Web-Based Development Workbench are not supported with XSA.

Note:
Be careful not to confuse SAP Web IDE for SAP HANA with SAP Web IDE. They are different tools with different use cases.

XSA Architecture

XSA's architecture is based on the following major technologies:

- Cloud Foundry
- Microservices Architecture
- Git/GitHub
- HANA Deployment Infrastructure (HDI)

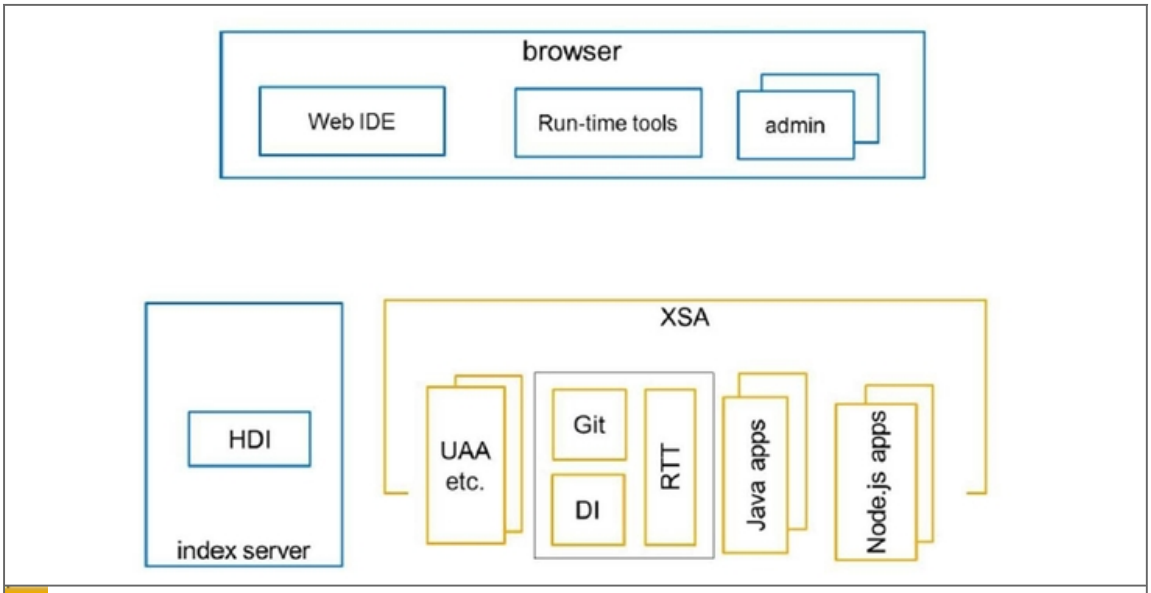


Figure 129: XSA Architecture

Cloud Foundry

Cloud Foundry is an open-source cloud platform-as-a-service (PaaS), on which developers can build, deploy, run, and scale applications on public and private cloud models. One of the key requirements for XSA was the desire to unify the architecture of solutions built in the cloud and on-premise. Cloud Foundry provides scalability options and flexible runtimes that are needed in the cloud environment. SAP HANA Cloud Platform is based on Cloud Foundry. For on-premise HANA we utilize only the basic aspects of Cloud Foundry. However, what is key is that we can now unify the developments of cloud and on-premise applications. This eliminates the wide gap that was present between the two with XS. One of the main advantages of Cloud Foundry is that we can build applications using multiple development languages. This means that we are not restricted to Javascript, but can also work with other languages including Java, node.JS, and C++.

Microservices Architecture

Another key innovation in XSA is the use of microservices architecture. Microservices architecture is a modern approach to building applications from individual modules that we combine. The modules can be developed in any language and are reusable across applications. Each module can be regarded as an individual service.

Microservices architecture allows us to follow the industry trend for building applications using the best tools and languages for the job (this is referred to as ‘bring your own language’).

Microservices enable better scaling and memory management, which means that each runtime runs in its own isolated container and can be individually tuned. It also means that if an individual service fails, it does not bring down the entire application. When an XSA application is called, the XSA routing service orchestrates the different runtime modules that make up the complete application.

Git/GitHub

Git is one of the industry-leading software version control frameworks. It is already used by a large number of developers to manage their software lifecycle. It can be used privately for internal projects or publicly for open-source projects where developers can contribute and share enhancements in a safe and controlled manner.

SAP have chosen Git as the new source code and design time object repository for XSA-based development. SAP Web IDE for SAP HANA has capabilities that allow the developer to interact directly with Git, in order to check in and check out development artifacts. We no longer store source code and other design time artifacts in the HANA repository, as we did with XS.

HDI

With XSA comes a brand new approach to deployment. We call this HANA Deployment Infrastructure (HDI).

At design time, we write code and also develop database objects in source Modules using Web IDE. A group of Modules makes up a Project. We then 'build' an application from the Modules of our project and the built application sits in its own Container. A container is an isolated unit that contains all of the run time objects that are needed in an application. We can redeploy the same application, with different features and enhancements, to other separate containers. This means that we can continue to run existing applications without disruption, while offering newer versions, or variations of the application that might not suit all consumers. We can deploy versions of the same applications to cloud or on-premise. This is a big change from how we deployed with XS. With XS we could only deploy one version of an application.

In summary, XSA provides customers with more choices of technologies, tools, and flexible deployment options than with XS. For these reasons, SAP strongly recommend that customers immediately use XSA, rather than continuing with XS.

Note:
To learn more about application development on SAP HANA, you may want to consider the course HA450, Application Development for SAP HANA .

XS to XSA Application Migration

Since HANA 1.0 SPS11 there are two very different application development and run time architectures available: XS Classic (XS) and XS Advanced (XSA). With SAP HANA 2.0, SAP strongly recommends to only use XSA.

But customers have already built many applications in XS, the original application development and run time environment introduced in the very early releases of SAP HANA. What should they do with these XS applications? Do they still run on SAP HANA 2.0?

Customers have three choices as to how they proceed:

- Continue to run the application under XS. SAP HANA 2.0 supports existing XS-based applications with no adjustments needed.
- Redevelop the application in XSA using newer tooling and multiple languages.
- Migrate the entire XS application to XSA.

The key motivation for the migration of XS based applications to XSA is to realign them to the more powerful and flexible architecture. Also, XSA is the only architecture that will be supported going forward.

XSA Advanced Migration Assistant

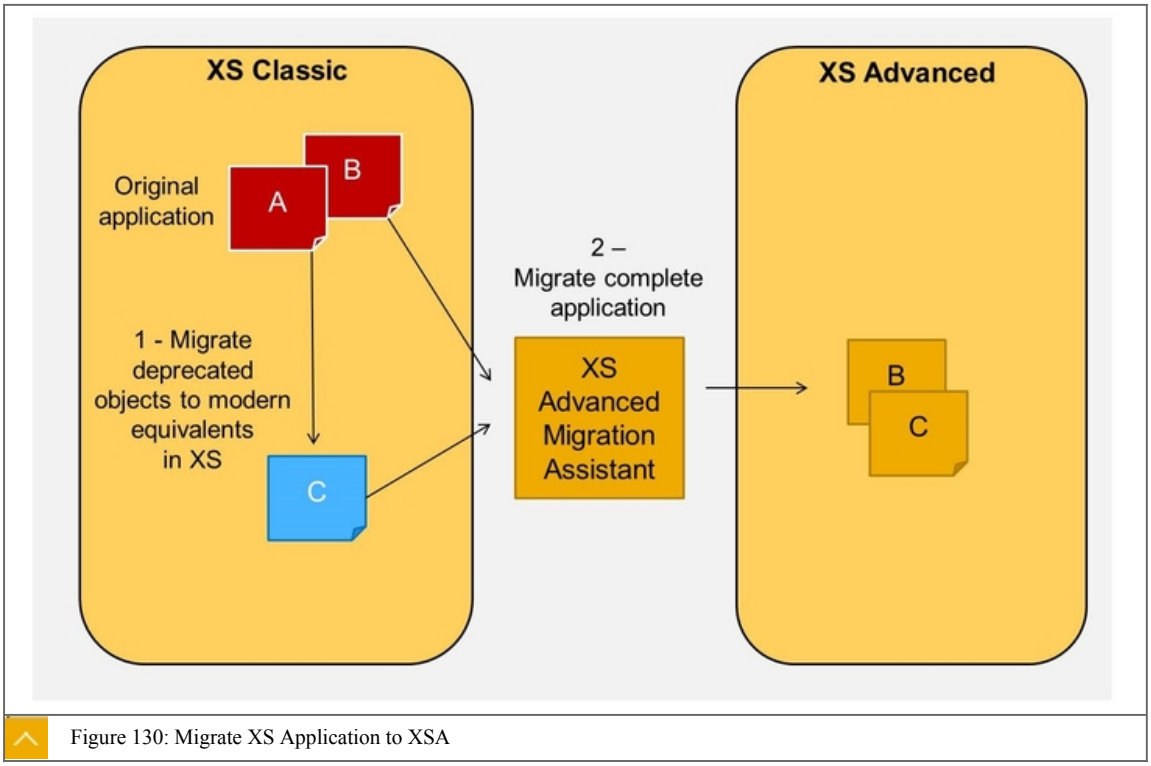


Figure 130: Migrate XS Application to XSA

To support the migration process, SAP provide tooling. The main tool is called **XS Advanced Migration Assistant** . This tool automates the migration of all objects that belong to an application so they become XSA objects.

However, the XS Advanced Migration Assistant is not able to migrate deprecated objects such as:

- Scripted Calculation Views
- Decision Tables
- Attribute Views
- Analytic Views
- XML based Analytic Privileges
- Application Function Library (AFL) models

You must first migrate those objects using the tooling that is found in the SAP HANA Studio. The migration of those objects essentially converts them to modern objects that are recognized in XSA and are supported going forward. For example, Scripted Calculation Views are converted to Table Functions. XSA does not recognize the deprecated objects.

Migration Process

Once the deprecated objects are migrated, you can then begin the migration of complete XS applications.

Migration is a very detailed process with many steps. The steps are summarized as follows:

1. Install XS Advanced Migration Assistant
2. Configure the connection to the XS classic and XS advanced systems
3. Migrate the XS classic application
4. Review the detailed report generated by the migration assistant
5. Upload and deploy the migrated application to XS advanced

To support the migration process, SAP deliver a real-life demonstration of the XS Advanced Migration Assistant using the SAP HANA Interactive Education (SHINE) demo application. We show you how to use the XS Advanced Migration Assistant to migrate one of the demo applications from XS classic to XS advanced. SAP have developed this SHINE capability in order to illustrate and explain the typical challenges you are likely to encounter when migrating XS classic applications.

Unit 5
Exercise 10

Operate an XS Advanced Native Application

Exercise Objectives

In this exercise you learn how to launch and operate an application that was built completely on SAP HANA that runs on XS Advanced.

Business Example

You would like to familiarize yourself with an application that was built completely on SAP HANA XSA.

Note:
In this exercise, when values include ##, replace these characters with your own student number.

1. Launch the SHINE XS Advanced application home page using the shortcut SHINE for XS Advanced.url that you will find in the HA100 folder, and log on using the details provided in the table, Logon Details .

   Table 14: Logon Details

Field	Value
User	**STUDENT##**
Password	**Training1**

2. Launch the Sales Dashboard and create a new sales order.
3. Select a business partner and create an order with two different products and different sales quantities.

Unit 5
Solution 10

Operate an XS Advanced Native Application

Exercise Objectives

In this exercise you learn how to launch and operate an application that was built completely on SAP HANA that runs on XS Advanced.

Business Example

You would like to familiarize yourself with an application that was built completely on SAP HANA XSA.

Note:
In this exercise, when values include ##, replace these characters with your own student number.

1. Launch the SHINE XS Advanced application home page using the shortcut SHINE for XS Advanced.url that you will find in the HA100 folder, and log on using the details provided in the table, Logon Details .

Table 14: Logon Details

Field	Value
User	**STUDENT##**
Password	**Training1**

a) On the Windows desktop, navigate to Favorites → HA100 → URLs.

b) Double click on the file SHINE for XS Advanced.url .

c) Log on using the details from the table, Logon Details .

d) Close the What's New dialog box using OK.

2. Launch the Sales Dashboard and create a new sales order.

a) Choose Sales Dashboard .

b) Close the dialog box using Continue .

c) Take a look at the content on the Overview tab and then choose the Details tab.

d) On the Details tab, choose New.

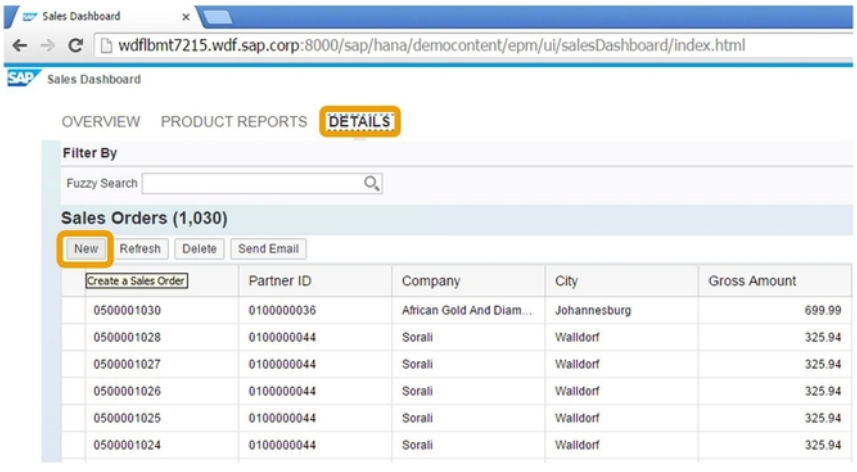


Figure 131: Create A Sales Order

3. Select a business partner and create an order with two different products and different sales quantities.

   a) From the Select a Business Partner drop down, select a business partner.

   b) From the Select a Product drop down, select a product.

   c) To add product quantities to the sales order, choose +.

   d) To save the order, choose Create .

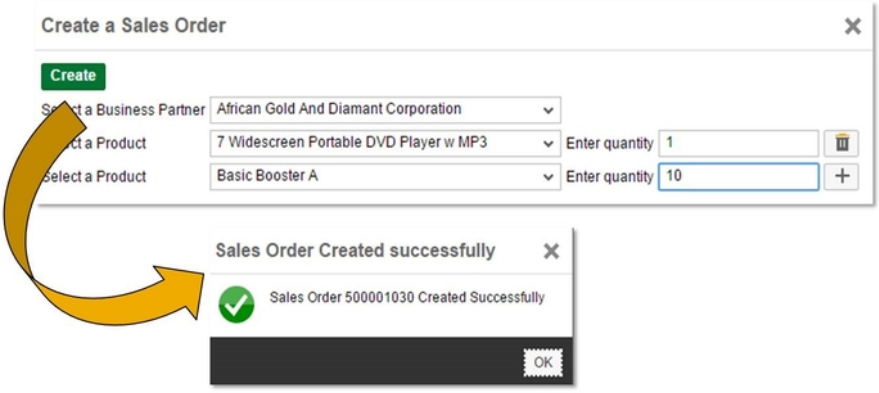


Figure 132: Sales Order Created

LESSON SUMMARY
You should now be able to:

- Describe the basics of native HANA applications

Unit 5

Learning Assessment

1. What are the database query languages used by SAP HANA?

   Choose the correct answers.

   - ☐ **A** ODBC
   - ☐ **B** JDBC
   - ☐ **C** MDX
   - ☐ **D** SQL
   - ☐ **E** ABAP

2. What are features of Business Suite on SAP HANA?

   Choose the correct answers.

   - ☐ **A** Embedded Analytics for operational analytics
   - ☐ **B** ABAP is replaced with JAVA
   - ☐ **C** Combine operational analytics and transaction processing
   - ☐ **D** Massive performance improvement

3. What are features of SAP S/4HANA?

   Choose the correct answers.

   - ☐ **A** Optimized for SAP HANA but maintains compatibility with other platforms
   - ☐ **B** Embedded Analytics
   - ☐ **C** Rewritten ABAP code optimized for SAP HANA
   - ☐ **D** SAP HANA Live provides real time operational analytics

4. To connect SAP Analysis Edition for Excel, you need to create a secure connection in SAP BusinessObjects Enterprise.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

5. Which of these does SAP HANA use to connect to BI tools?

   Choose the correct answers.

   - ☐ **A** IDOC
   - ☐ **B** ODBC
   - ☐ **C** JDBC
   - ☐ **D** ODBO

6. In which key areas can we expect there to be improvements when running SAP BW on SAP HANA?

   Choose the correct answers.

   - ☐ **A** Tighter security
   - ☐ **B** Data loading performance
   - ☐ **C** Superior data quality
   - ☐ **D** Reporting performance

7. In SAP BW, when we refer to mixed modeling, what do we mean?

   Choose the correct answer.

   - ☐ **A** Integration of classic BW models such as InfoSets or InfoCubes with SAP HANA Calculation views
   - ☐ **B** Integration of SAP HANA optimized BW models such as ADSOs or OpenODS views with SAP HANA Calculation views
   - ☐ **C** Integration of models from BW on non-HANA databases combined with SAP HANA optimized BW models

8. Where will you find the Data Distribution Optimizer?

   Choose the correct answer.

   - ☐ **A** SAP HANA Studio
   - ☐ **B** SAP BW/4HANA
   - ☐ **C** SAP HANA Data Warehousing foundation

9. Which interface client is used for XSA application development?

   Choose the correct answer.

   - ☐ **A** SAP HANA Web-based Workbench
   - ☐ **B** SAP Web IDE for SAP HANA
   - ☐ **C** SAP HANA Studio
   - ☐ **D** SAP Web IDE

10. Applications developed with XS must be migrated to XSA before they can run in SAP HANA 2.0.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

Unit 5

Learning Assessment - Answers

1. What are the database query languages used by SAP HANA?

   Choose the correct answers.

   - [ ] **A** ODBC
   - [ ] **B** JDBC
   - [X] **C** MDX
   - [X] **D** SQL
   - [ ] **E** ABAP

   You are correct! ODBC, JDBC and ABAP are not database query languages. MDX and SQL are database query languages supported by SAP HANA. Refer to HA100, Unit 5, Powering SAP Applications with SAP HANA for details.

2. What are features of Business Suite on SAP HANA?

   Choose the correct answers.

   - [ ] **A** Embedded Analytics for operational analytics
   - [ ] **B** ABAP is replaced with JAVA
   - [X] **C** Combine operational analytics and transaction processing
   - [X] **D** Massive performance improvement

   You are correct! Embedded Analytics are only relevant for SAP S/4HANA. Business Suite on HANA uses SAP HANA Live. ABAP is still used. Business Suite on SAP HANA now combines operational analytics and transaction processing and you will get a massive boost in performance. Refer to HA100, Unit 5, Powering Applications with SAP HANA , for details.

3. What are features of SAP S/4HANA?

   Choose the correct answers.

   [ ] **A** Optimized for SAP HANA but maintains compatibility with other platforms

   [X] **B** Embedded Analytics

   [X] **C** Rewritten ABAP code optimized for SAP HANA

   [ ] **D** SAP HANA Live provides real time operational analytics

   You are correct! SAP S/4HANA runs only on the SAP HANA platform. ABAP code has been completely rewritten so it works optimally with SAP HANA. SAP HANA Live is relevant only for Business Suite on SAP HANA. S/4HANA uses Embedded Analytics to provide real time operational analytics. Refer to HA100, Unit 5, Powering Applications with SAP HANA, for details.

4. To connect SAP Analysis Edition for Excel, you need to create a secure connection in SAP BusinessObjects Enterprise.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! Although an SAP BusinessObjects Enterprise secure connection can be used, you can also connect directly to SAP HANA without needing this. Refer to HA100. Unit 5, Powering Applications with SAP HANA , for details.

5. Which of these does SAP HANA use to connect to BI tools?

   Choose the correct answers.

   [ ] **A** IDOC

   [X] **B** ODBC

   [X] **C** JDBC

   [X] **D** ODBO

   You are correct! IDOC is an SAP NetWeaver technology used to pass data between SAP systems so is not valid here. Refer to HA100, Unit 5, Powering Applications with SAP HANA, for details.

6. In which key areas can we expect there to be improvements when running SAP BW on SAP HANA?

   Choose the correct answers.

   - [ ] **A** Tighter security
   - [X] **B** Data loading performance
   - [ ] **C** Superior data quality
   - [X] **D** Reporting performance

   You are correct! Security and data quality are not improved when running BW on SAP HANA but loading and reporting are impacted greatly in a very positive way. Refer to HA100, Unit 5, Powering Applications with SAP HANA , for details.

7. In SAP BW, when we refer to mixed modeling, what do we mean?

   Choose the correct answer.

   - [ ] **A** Integration of classic BW models such as InfoSets or InfoCubes with SAP HANA Calculation views
   - [X] **B** Integration of SAP HANA optimized BW models such as ADSOs or OpenODS views with SAP HANA Calculation views
   - [ ] **C** Integration of models from BW on non-HANA databases combined with SAP HANA optimized BW models

   You are correct! Mixed modeling integrates SAP HANA Calculation views and SAP HANA optimized BW models such as DSOs. Refer to HA100, Unit 5, Powering Applications with SAP HANA, for details.

8. Where will you find the Data Distribution Optimizer?

   Choose the correct answer.

   - [ ] **A** SAP HANA Studio
   - [ ] **B** SAP BW/4HANA
   - [X] **C** SAP HANA Data Warehousing foundation

   You are correct! Data Distribution Optimizer is key tool of Data Warehouse foundation and it helps you to manage data locality by ensuring tables that are frequently processed together, stay together and are not split up across nodes in a scale–out landscape : Refer to HA100 Unit 5, Powering Applications with SAP HANA, for details.

9. Which interface client is used for XSA application development?

   Choose the correct answer.

   - [ ] **A** SAP HANA Web-based Workbench
   - [X] **B** SAP Web IDE for SAP HANA
   - [ ] **C** SAP HANA Studio
   - [ ] **D** SAP Web IDE

   You are correct! Neither SAP HANA Studio nor Web Based Workbench are used with XSA, as they are used with XS application development only. Web IDE is not used with SAP HANA, as it is used for SAP Fiori and SAPUI5 application development. Refer to HA100, Unit 5, Powering Applications with SAP HANA , for details.

10. Applications developed with XS must be migrated to XSA before they can run in SAP HANA 2.0.

    Determine whether this statement is true or false.

    - [ ] True
    - [X] False

    You are correct! SAP HANA 2.0 supports XS applications without adjustment; they do not need to be migrated to XSA. Refer to HA100, Unit 5, Powering Applications with SAP HANA, for details.

UNIT 6

Appendix: Continuing your SAP HANA Learning Journey

UNIT OBJECTIVES

- Develop your SAP HANA knowledge

Unit 6
Lesson 1

Developing your SAP HANA Knowledge

LESSON OVERVIEW
In this lesson we will introduce the various follow-on courses and other educational resources available to help you continue your SAP HANA learning journey.

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Develop your SAP HANA knowledge

Further SAP HANA courses

SAP HANA education content is organized into five tracks, which align to the skills that are in demand in the marketplace. These tracks are as follows:

- SAP HANA Modeling
- SAP HANA Administration and Operations
- SAP HANA Application Development
- SAP HANA Data Provisioning
- SAP Business Warehouse powered by SAP HANA

The key courses for each track are included below and are intended to provide you with suggestions for your next SAP HANA course.

SAP HANA Modeling

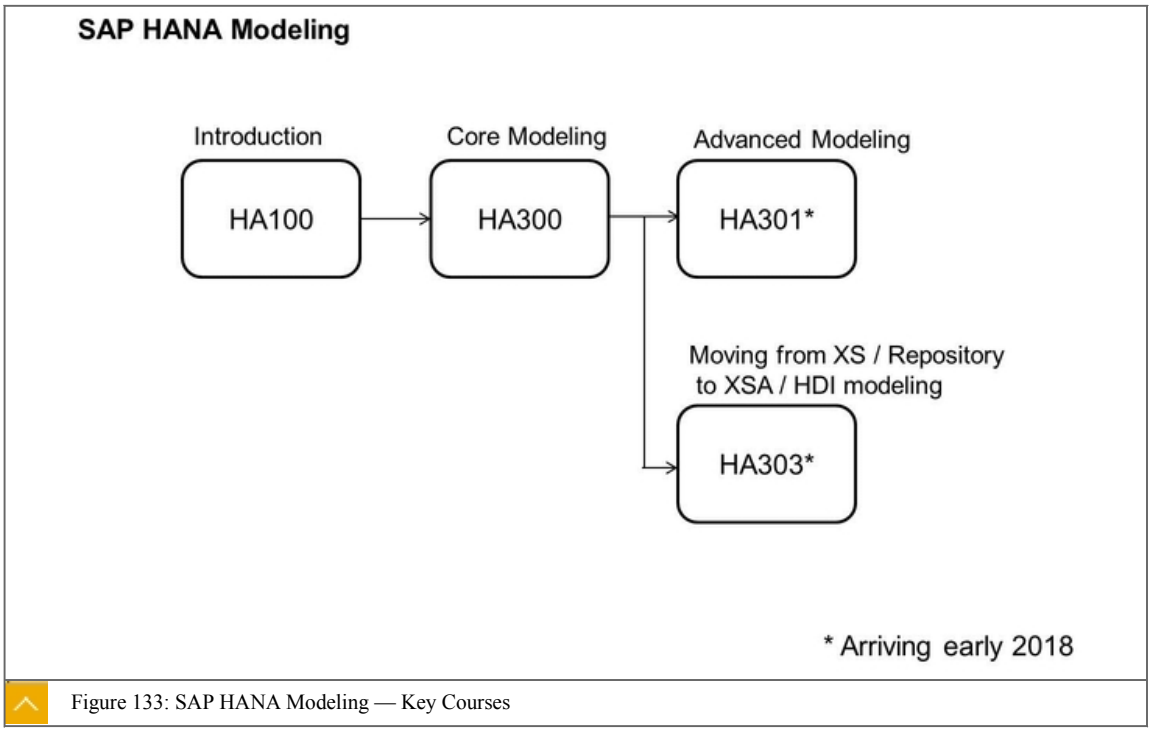


Figure 133: SAP HANA Modeling — Key Courses

SAP HANA Administration and Operations

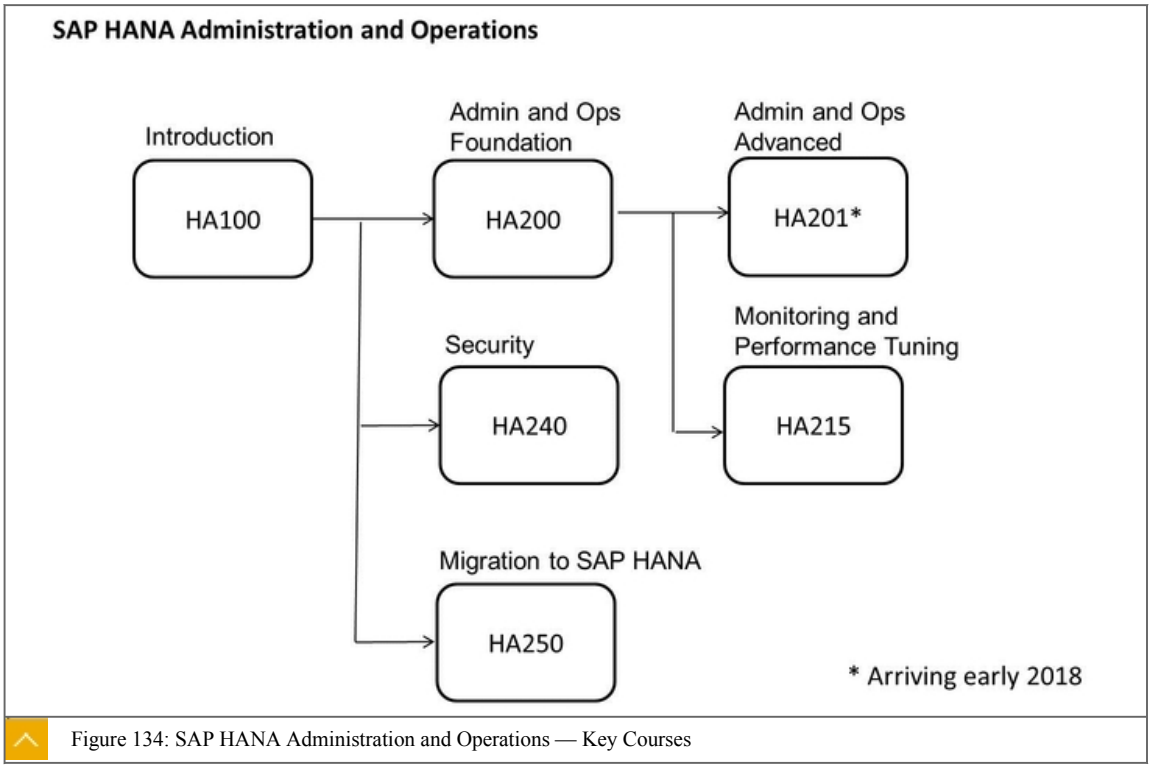


Figure 134: SAP HANA Administration and Operations — Key Courses

SAP HANA Development

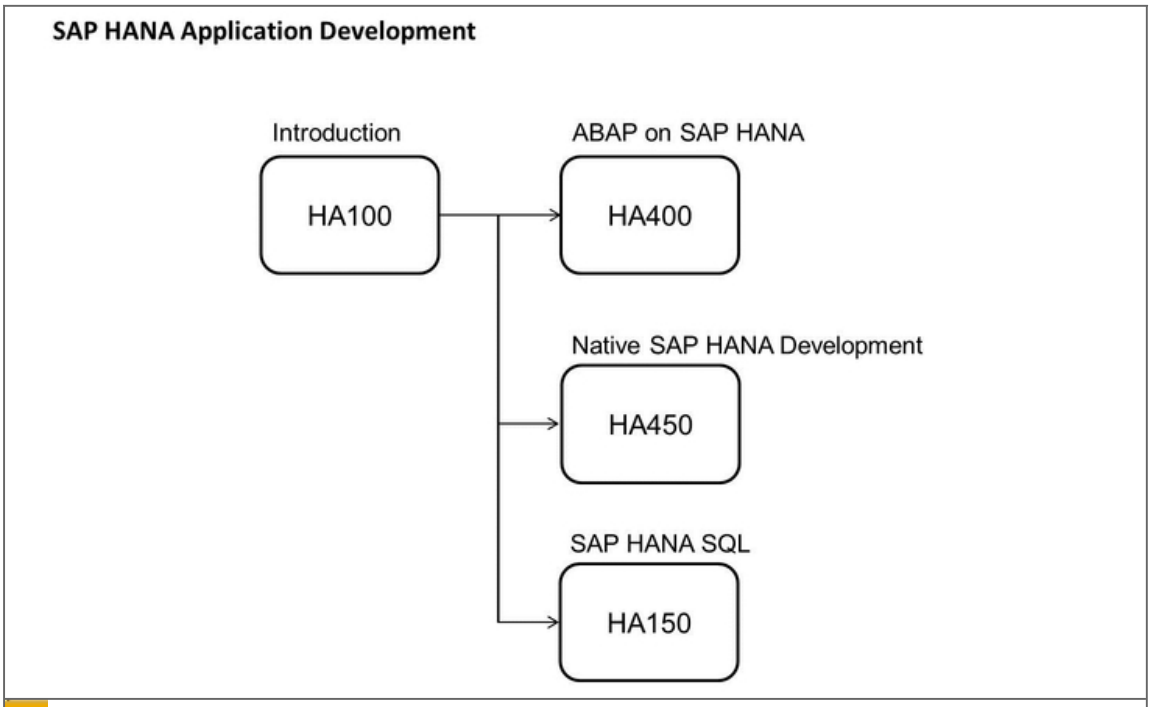


Figure 135: SAP HANA Application Development — Key Courses

SAP HANA Data Provisioning

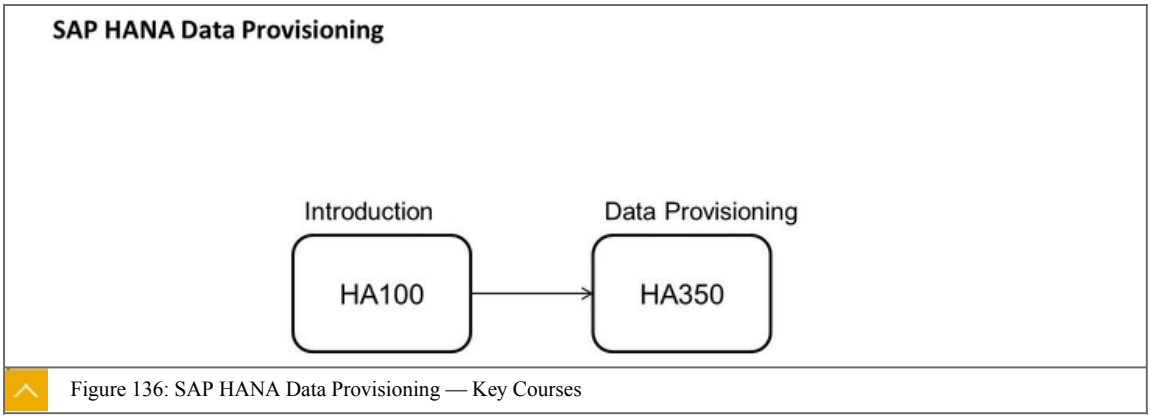


Figure 136: SAP HANA Data Provisioning — Key Courses

SAP BW powered by SAP HANA and SAP BW/4HANA

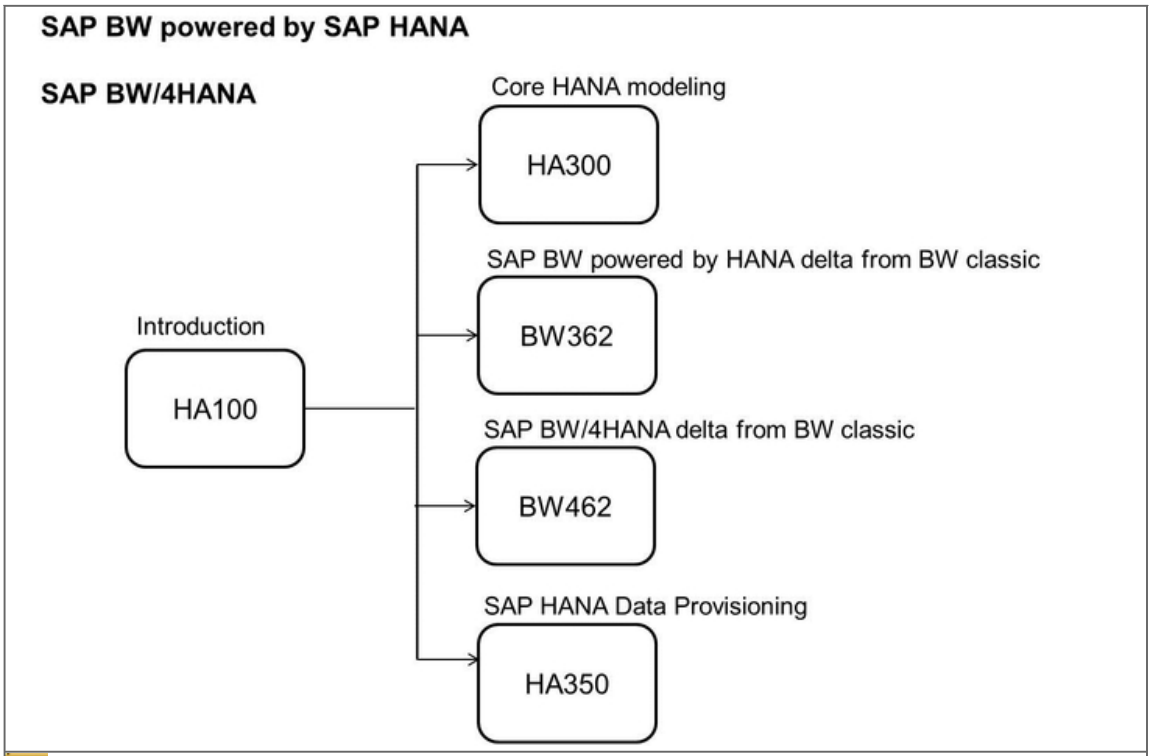


Figure 137: SAP BW powered by SAP HANA — Key Courses

Learning Journeys

For each track, SAP provide an **SAP Learning Journey** , which helps you identify which piece of education content you should follow and in the sequence that is most sensible. These include not only the courses described above, but also other types of helpful education content such as OpenSAP workshops and certifications to create a complete end-to-end journey.

The SAP Learning Journeys are found at: http://help.sap.com/learningjourneys .

Consumption Options for SAP HANA Courses

There are three modes for consumption of SAP HANA courses. They are as follows:

- Classroom

  These are public scheduled events at an SAP Training Center. Alternatively, in most regions, you can have the class delivered privately on site.

- Virtual Live Class (VLC)

  These are public scheduled events where all students are connected via the internet, with full two-way communication with the instructor. You are sent the full course materials in advance as well as the SAP HANA training system access details. You will be part of a live event with the same duration as a classroom event covering exactly the same content. You can easily ask questions using voice or chat pods.

- E-learning

  E-learning presents exactly the same materials as the other consumption options, but you take the self-study courses in your own time. You access the e-learning version of all SAP

courses by subscribing to SAP Learning Hub. You then have access to all slides and demos (which are narrated by a speaker), on-line quizzes, plus the course handbook.

Note:
When you see references to ILT, this means Instructor Lead Training which is a live event run by an instructor. Classroom and VLC events are both ILT. E-learning is not ILT.

Please refer to the SAP Education shop at training.sap.com for the very latest information on all courses, locations, prices, and availability.

SAP Learning Hub

SAP Learning Hub is SAP Education's **cloud** solution and provides unlimited access to all SAP education content enhanced by social learning and peer collaboration. SAP Learning Hub is a subscription based service. SAP Learning Hub consists of three key components:

- Learning Content — All SAP courses in the global portfolio are included (including this one)
- Learning Rooms — Spaces for collaboration with SAP experts and other learners
- Access to official SAP Training Systems — Fully configured training servers where you can work through all course exercises

Let's describe each component in a little more detail.

Learning Content

Learning content includes unlimited access to SAP e-learning courses where you have the same content as the classroom courses. You will see the same slides and demos with expert recorded narration and also the course handbooks are available to you, plus on-line assessments to help you gauge your progress. You can work on as many courses as you wish, sequentially or simultaneously. You can dive into any topics in any course as you need to meet your on-demand learning requirements.

Refer to the course maps provided earlier for a list of SAP HANA related courses that you will find in SAP Learning Hub content.

SAP Learning Rooms

SAP Learning Rooms are SAP instructor-led collaborative communities that support you in your learning journey. In the Learning Rooms you can do the following:

- Ask questions anytime to SAP instructors and also review the archive of previous questions
- Watch additional videos developed and posted by the instructors and other SAP experts
- Attend regular live sessions hosted by experts covering key topics, and ask your questions
- Catch-up with the live session recordings if you missed any
- Read blogs from experts
- Check your HANA knowledge using our exam preparation aids and sample questions

SAP Learning Rooms are ideal for students who are preparing for certification. There are assessments and tips available to ensure that you are ready for the exams.

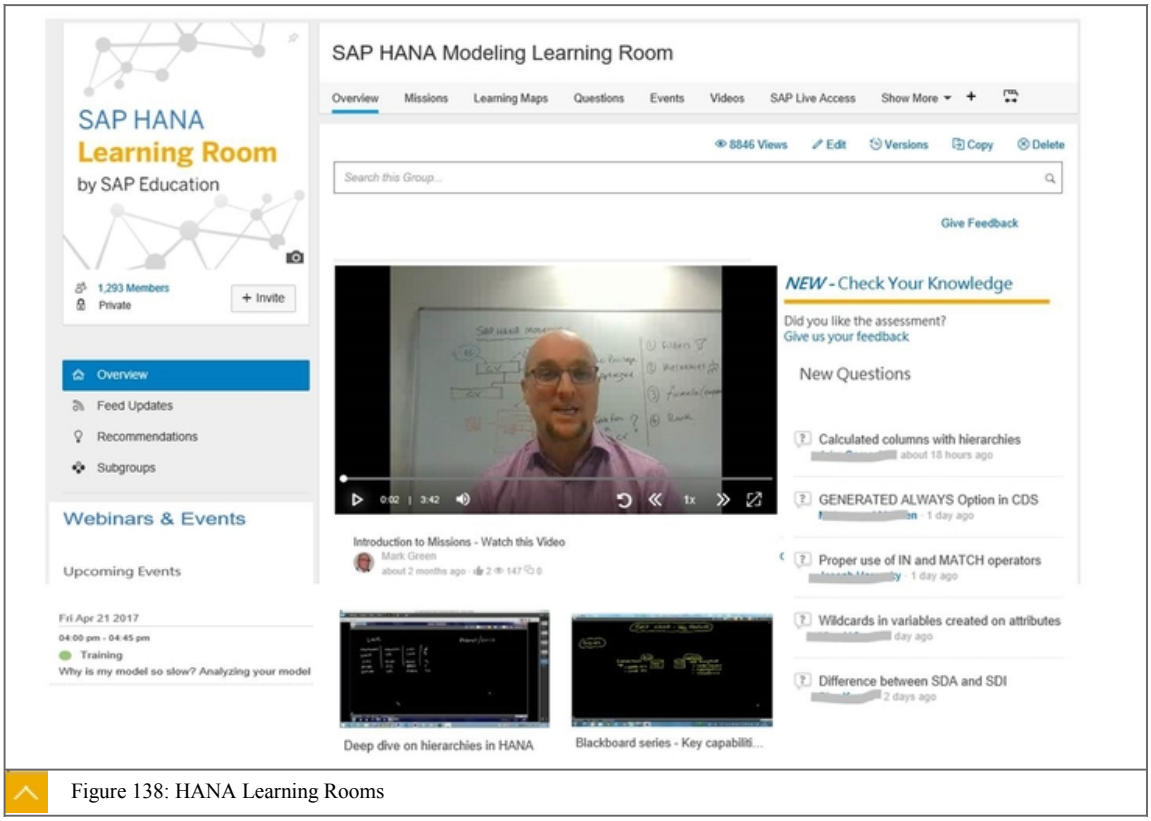


Figure 138: HANA Learning Rooms

There are many different SAP Learning Rooms and they align to popular SAP topics, such as SAP HANA modeling and SAP HANA administration. SAP Learning Rooms are available as part of a subscription to SAP Learning Hub at no additional cost. Learning Hub subscribers can join as many Learning Rooms as they wish. There are currently almost 100 Learning Rooms available and more are being added all the time.

SAP HANA Learning Rooms available in the SAP Learning Hub

Learning Room Title	Description	Learning Objectives
SAP HANA Implementation & Modeling	This room will provide you with a collaborative environment to help you work towards taking the SAP Certified Application Associate - SAP HANA certification exam. Of course, if you don't plan to take the certification, but still wish to learn more about modeling with SAP HANA then this room is for you	• Develop a broad view of SAP HANA, it's position as a platform, the architecture, and development tools • Develop detailed knowledge in the area of SAP HANA Implementation, Modelling, and Data Provisioning. • Prepare yourself for SAP HANA certification: C_HANAIMPXXX - SAP Certified Application Associate - SAP HANA
ABAP for SAP HANA	This room will provide you with a collaborative environment to help you work towards taking the ABAP for HANA specialist certification exam. Of course, if you don't plan to take the certification, but still wish to learn more about ABAP for HANA, then this room is for you. With expert instructors on hand to answer any questions you have, appropriate self-study material, extra content on the most important topic areas, and live online sessions where you will have the chance to interact with an instructor and each other, this room is a great way to work towards certification.	• The ABAP for HANA Certification room (E_HANAAWXXX - SAP Certified Development Specialist (Edition 2015) - ABAP for SAP HANA) is intended to guide learners through the topic areas needed for the ABAP for HANA Specialist certification exam.
SAP HANA Administration and Operations	This room will provide you with a collaborative environment to help you work towards taking the SAP Certified Technology Associate - SAP HANA certification exam. Of course, if you don't plan to take this certification, but still want to learn more about SAP HANA Technology, then this Learning Room is for you.	• Prepare for the Certification exam: C_HANATECXXX - SAP Certified Technology Associate - SAP HANA
SAP BW powered by SAP HANA	This room will provide you with a collaborative environment to help you work towards taking the SAP Certified Application Specialist - SAP BW powered by SAP HANA exam. Of course, if you don't plan to take this certification, but still want to learn more about SAP BW on SAP HANA, then this Learning Room is for you.	• Gain a fundamental understanding of the possibilities and the positioning of SAP BW powered by SAP HANA, with an ultimate objective to apply this knowledge as a solution consultant in a team setting. • Prepare for the Certification exam: E_HANABWXXX -

Figure 139: SAP HANA Recommended Learning Rooms

SAP HANA Training System Access

A subscription to SAP Learning Hub provides exclusive rights to access the training systems that we use for all SAP HANA classes. To get started you need to purchase an access voucher. Each system is a fully-configured SAP HANA landscape and is not shared by others. It is a complete SAP HANA landscape just for your use. SAP Live Access is ideal when you want to work through all the exercises in a class, or even just to explore or perhaps prepare for certification.

SAP HANA – Live Training System Access

Course Code	Course Title	Target Audience
HA100	SAP HANA – Introduction	Application Consultants Project team members
HA150	SQL Script: Basics and Advanced for SAP HANA	Application Consultants, Development Consultants
HA200	SAP HANA – Installation & Operations	Technology Consultants System Administrators
HA215	SAP HANA – Monitoring and Performance Tuning	Technology Consultants System Administrators
HA240	SAP HANA Authorizations, Scenarios & Security Requirements	Technology Consultants System Administrators
HA250	SAP HANA – Migration to SAP HANA	Technology Consultants System Administrators
HA300	SAP HANA – Implementation & Modeling	Application Consultants Project team members Data Modeling Consultants
HA400	ABAP Programming for SAP HANA	Developers with experience in ABAP programming
HA450	Application Development for SAP HANA	Developers Development consultants
BW362	SAP BW on SAP HANA	Application consultants

Figure 140: SAP HANA Training System Access

We have fully-configured training systems in place to support all of these SAP HANA courses for your own personal access.

SAP HANA Interactive Education (SHINE)

SAP HANA Interactive Education (SHINE) is a bundle of SAP-supplied showcase applications that run on SAP HANA. Your application developers can use this demo model to discover how to build native SAP HANA applications.

The applications include text, spatial, and other basic and advanced features. The idea is that you can learn from these working applications rather than struggle yourself to get something working.

SHINE is downloaded. There is a version for XS as well as a version for XSA that developers can dismantle, to investigate how SAP have created the showcase applications. SHINE comes with its own data generator and documentation that describes how the applications are put together.

SHINE

Figure 141: SHINE

Note:
SAP HANA SHINE is a great resource for anyone who wants to learn HANA modeling and native HANA application development. It comes with ready-to-go, well-built, interesting applications and it is completely free of charge.

Do not install SHINE on your productive SAP HANA system.

LESSON SUMMARY
You should now be able to:

- Develop your SAP HANA knowledge

Glossary

AFL
Application Function Library is a repository of SAP supplied modules for predictive analysis and building business applications

Analytic Privilege
Object to define the data access rules for users

Analytic View
Modeling object used to define a star schema based data model, but is now replaced by the calculation view.

Attribute
A column used to bring meaning to a measure, for example to aggregate by country or year and used to drill-down.

Attribute View
Modeling object used to define dimensions and their attributes that can be used stand-alone or integrated in a star schema, but is now replaced by the calculation view.

BICS
Business Intelligence Consumer Service is an SAP developed access technology used to tightly integrate SAP data sources such as SAP HANA with SAP analysis applications such as Design Studio for Excel.

Build
Generate the run-time objects from the design-time definitions for one of more objects at a time.

BWA
Business Warehouse Accelerator is an optional BW add-on that speeds up reporting from BW by using in-memory technology. It went on to become the origins of SAP HANA technology.

Calculation View
Core modeling object used to define data model of any type including dimension, cube, and cube star schema, and replaces attribute and analytic views.

CDS
Core Data Services is a scripting language used to define the persistence layer and also the data modeling layer with the ability to add rich semantics to provide technical and business meaning to data.

Cloud Foundry
Open source, industry standard cloud platform used to build, deploy and run and scale applications now adopted by SAP HANA for XSA developments.

Container
Collector of all database run-time objects used by an application. It is a logical layer that sits above the physical database schema where the actual database objects reside, used only by XSA applications.

Cube
Multi-dimensional data set used in slice and dice analysis.

Data Warehouse Foundation
Toolset that can be installed in SAP HANA to provide developers with tools to build a data warehouse completely on SAP HANA without the need for BW.

Delivery Unit
Used to transport development artifacts from one HANA system to another, used only in classic XS developments.

Dimension
Collection of attributes that belong together (e.g. product color, weight, price) used to provide meaning to measures and can be shared across many calculation views.

Extended Storage
An optional disk based data store that is fully integrated into the SAP HANA database and often used to off-load less important data from memory (hot) to a warm store.

Flowgraph
Object used to define an SDI data flow using a graphical editor, and also used to define a predictive analysis model.

Function
Custom read-only SQL based script used to either return a tabular result set that can be used as a data source in a calculation view (table function), or to return a single value to be used as an input parameter (scalar function).

Gerrit
Web based code review tool that integrated with Git that supports a workflow approach to collaborative

code review and approval and used with SAP HANA XSA development projects.

Git
Industry standard, and very popular source code version control solution used by SAP HANA XSA developers to easily share code and modeling artifacts, and replaces the internal source code versioning with SAP HANA XS..

Graph
A model that represent entities that are best described in a network, such as a social network (Facebook)

HDI
HANA Deployment Infrastructure — The 'all or nothing' deployment infrastructure used by XSA applications.

MDX
Multi Dimensional Expression language was developed by Microsoft to provide an alternative to SQL for queries that access multi-dimensional data sources to be written in a less complex way and with dedicated syntax.

Measure
A numerical value that represents money, quantities or general numbers that can be aggregated and used in calculations.

MTA
Multi Target Application is an SAP HANA single application that can be deployed both on premise and in the cloud without adjustment of the code.

OData
Open Data Protocol is a poplar and open standard introduced by Microsoft and supported by XS and XSA that allows a developer to expose a HANA table or view to an application by defining reusable query logic.

ODBC
Open Database Connectivity is a protocol originally developed by Microsoft to provide connectivity between applications to any relational database. It is a very popular method of connecting application to SAP HANA.

ODBO
OLE DB for OLAP was introduced by Microsoft to provide a connectivity interface between slice and dice analysis tools such as XL pivot tables, and multi-dimensional data models, such as cube calculation views

Organization
Used in XSA developments to group Spaces that enable sharing of resources among developers./

Package
Used to organize development artifacts in a hierarchical manner, used only in classic XS developments.

PAL
Predictive Analysis Library is a repository of SAP supplied, ready-to-use algorithms and functions meant to be used in predictive analysis models.

Persistence layer
The layer built using database objects that physically store data as opposed to the virtual layer that sits above it.

Procedure
Customer-SQL-based script used to define reusable processing logic that can be used universally across HANA to read and also write to the database. Sometimes called a stored procedure.

Project
Collector of all design-time artifacts used in an application.

Role
A collection of privileges that can be assigned to another role or a user

SAP Vora
Big Data integrator based on Apache Spark used to add a processing layer on top of any Big Data store in order to combine Big Data with HANA data.

SDA
Smart Data Access — In-built SAP HANA technology to expose remote data source as virtual tables in the SAP HANA database.

SDI
Smart Data Integration — In-built component of SAP HANA used to integrate and harmonize data from single or multiple sources of any type, in batch or real time.

SDQ

Smart Data Quality — In-built component of SAP HANA used to improve data quality by enrichment and cleansing of data being loaded through SDI.

SHINE
SAP HANA Interactive Education is a prebuilt model application supplied and maintained by SAP to showcase what can be built in SAP HANA that developers can study and learn from.

SLT
SAP Landscape Transformation Server is SAP's popular NetWeaver replicator tool used to move data from one system to another (including SAP HANA) in real time, also used in SAP S/4HANA Central Finance solution.

SP
Service Pack relates to an individual component of SAP HANA, such as SAP HANA Live tools SP 1.0

Space
Used in XSA developments to provide different collections of shared resources to various projects.

SPS
Support Pack Stack is a two-digit number that identifies the release of SAP HANA, e.g. SAP HANA SPS12.

SQL
Structured Query Language is used to read and write from relational databases. It is the world's most popular language for doing this.

Studio
Interface used by developers and modelers and administrators for building XS data models and applications.

Synonym
An alias that is used to provide access to a database object that resides outside the local container of an XSA application, mostly used to access shared objects that should not be part of an application ownership.

Web IDE for SAP HANA
Web Interface used by developers and modelers for building XSA data models and applications.

Web-based Development Workbench
Web interface used by developers and modelers to build XS data models and applications, and a lightweight alternative to Studio.

Workspace
A design time area used by a developer to build applications and models.

XS
Extended Application Services is SAP's first generation framework that provides all components needed to build and run applications completely in SAP HANA.

XSA
Extended Application Services — Advanced is SAP's second generation framework that provides all components needed to build and run applications completely in SAP HANA and now follows common cloud standards for more flexible deployment options.