HA100

SAP HANA - 360° Introduction

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 18
Course Duration: 3 Day(s)
Material Number: 50159944

SAP Copyrights, Trademarks and Disclaimers

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
- Describe the key technologies of SAP HANA
- Understand deployment options of SAP HANA
- Describe the Key Roles of an SAP HANA Implementation

Unit 1
Lesson 1

Understanding the Need for a Modern Digital Platform

LESSON OVERVIEW

This lesson will help you to appreciate the demands being placed on organizations caused by an increasingly digital world. The lesson will also help you to understand why business software needs to keep up with the fast changing technology.

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Understand the need for a modern digital platform

New Opportunities in a Digital World

In our changing world we are more connected than ever before, with more valuable data being generated every day by organizations and individuals. More and more existing services are moving online and new digital businesses are appearing. 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.

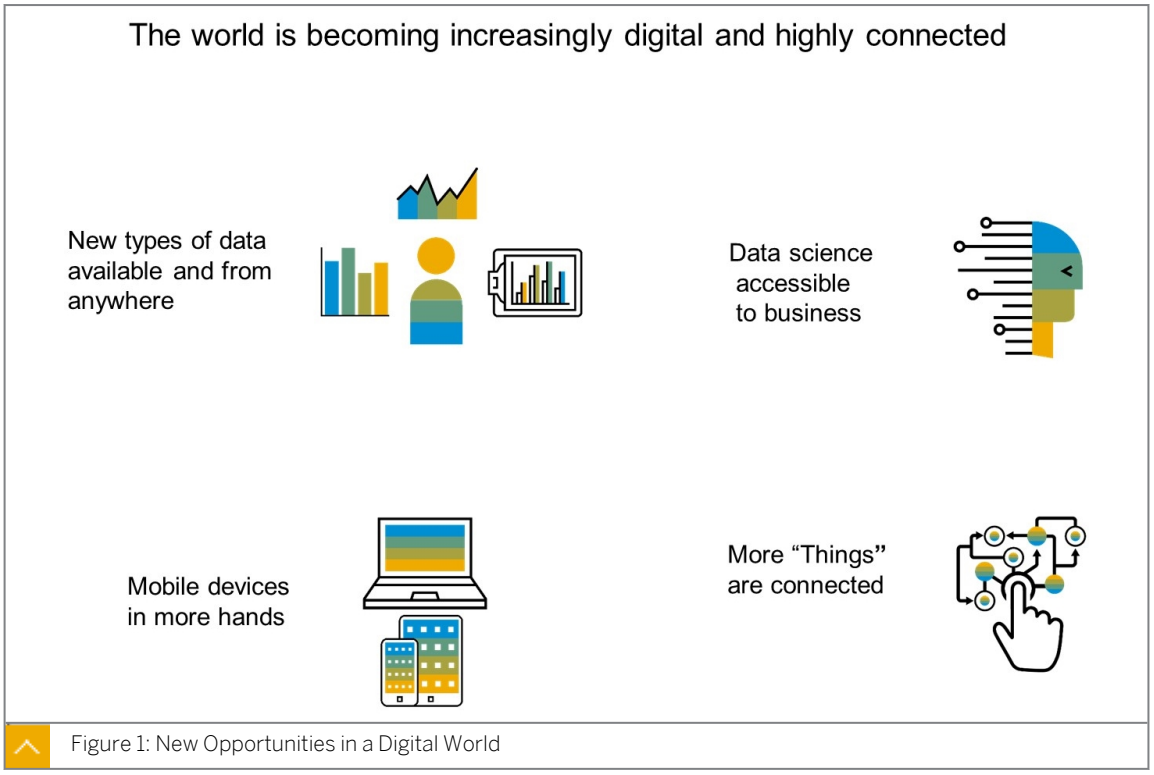


Figure 1: New Opportunities in a Digital World

The exponential growth of mobile devices, social media and the huge data volumes they generate has transformed the way we live and work.

Most companies report that the majority of their people use mobile devices in their roles to handle everything from e-mail to project management to content creation.

But it is not just people who are connected. We are seeing a dramatic increase in the number of devices that have become “intelligent” by including internet connectivity built right inside. We refer to this trend as the Internet of Things (IoT). Nowadays, even devices can communicate with each other in real-time to create intelligent and responsive environments.

New types of data are available. As well as traditional, structured columns and rows that represent master data and transactions, we now have access to other types of data such as unstructured text data, spatial data and graph data. We can now develop applications that combine these data types to provide even more insight.

Until recently, data science was out of reach for most organizations who did not employ highly skilled personnel to work in this area. Today, data science is within the reach of business users who can use simple tools to develop powerful data models that reveal valuable insights into their vast amounts of data.

These are exciting times and there are unprecedented opportunities for all organizations to grow their businesses exploiting the latest technologies.

IT Faces Challenges with Current Landscapes

Typical IT landscapes have developed over time into multiple complex arrangements of purchased, acquired with developed applications, 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 application development languages. To pull together, these different applications, organizations have created interfaces between systems. But these interfaces require constant maintenance as individual systems evolve. This in turn adds to the complexity.

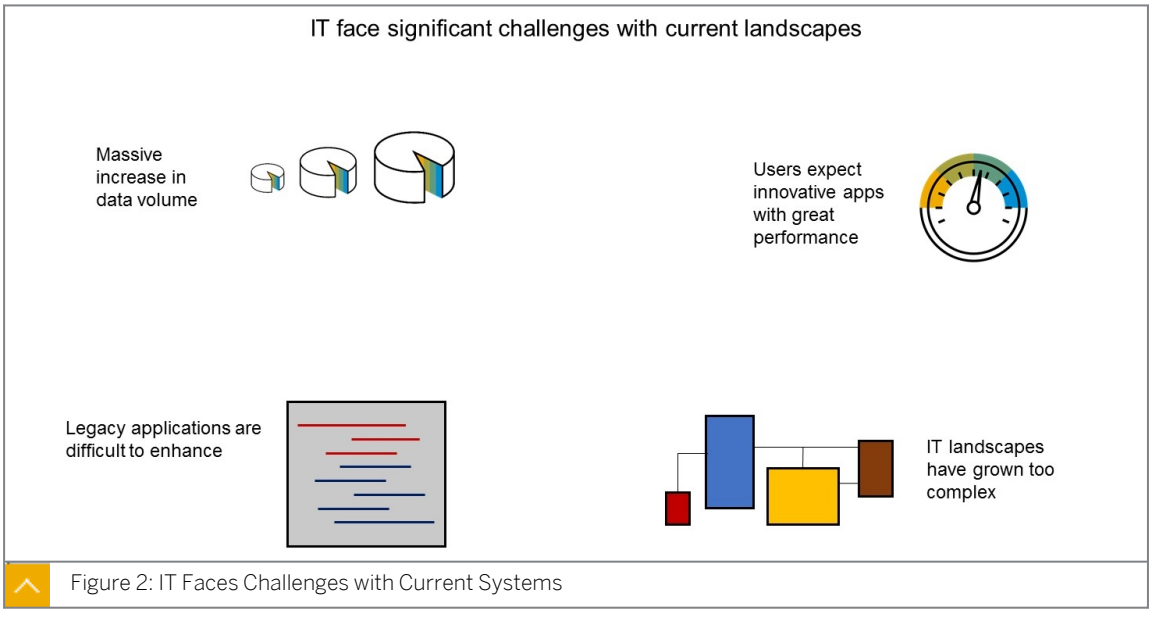


Figure 2: IT Faces Challenges with Current Systems

Integrating mass data storage solutions into a landscape is a challenge because legacy applications were not designed to manage data stored in various online tiers of various temperatures. The applications simply expected the data to be off-loaded to archives and out of sight. They do not know how to access cloud data or data lakes.

Consumerization is driving expectations of what business IT should offer its users. As business users become familiar with smart, instant-response consumer apps in their private lives, they also demand real-time, innovative applications in their jobs. These applications are needed to enable deeper insight and provide high-value decision support to enable them to perform well in their roles. Users also expect to be able to carry our their duties using mobile devices and not be tied to desktops.

Organizations have developed applications that have been extended and modified over many years. These applications are increasingly difficult to enhance as the code becomes more complex. A business user asking for a small adjustment to an application, can be a major undertaking.

For many organization however, their current business systems cannot cope with the increasing demands of the highly connected digital world that is generating data at scales we have never seen before. 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 be built on the latest technologies.

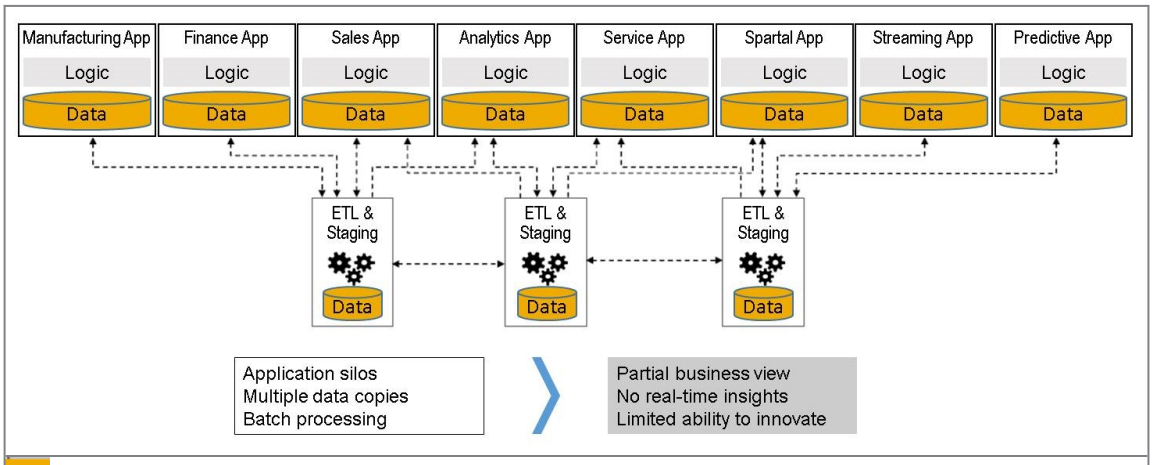


Figure 3: Siloes of Data

One of the biggest issues is the inefficiency of the shift and lift' approach present in many IT landscapes. In the past, it was necessary to move data away from transactional applications into dedicated systems so that data could be harmonized and cleaned. As a result, multiple copies of the data are created. But even worse, there is latency between the systems while data is transferred. This means that we do not have a consistent picture of data across the enterprise at any moment.

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 applications and developing new ones.

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

- Describe how SAP HANA powers a digital platform

Exploiting Technological Advancement

Some of the questions that may arise 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 technology. SAP HANA was not built by taking existing software and building on top of it. This redevelopment was undertaken to ensure that it is able to handle such an ambitious challenge and provide a next generation platform that aligns with new approaches.

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 CPU was if the data could not reach it quickly.

In recent years, the cost of memory has fallen and continues to fall year-on-year. Hardware vendors are now shipping huge amounts of memory that is affordable. Memory can now scale up to many terabytes with up to 128 TB memory now achievable.

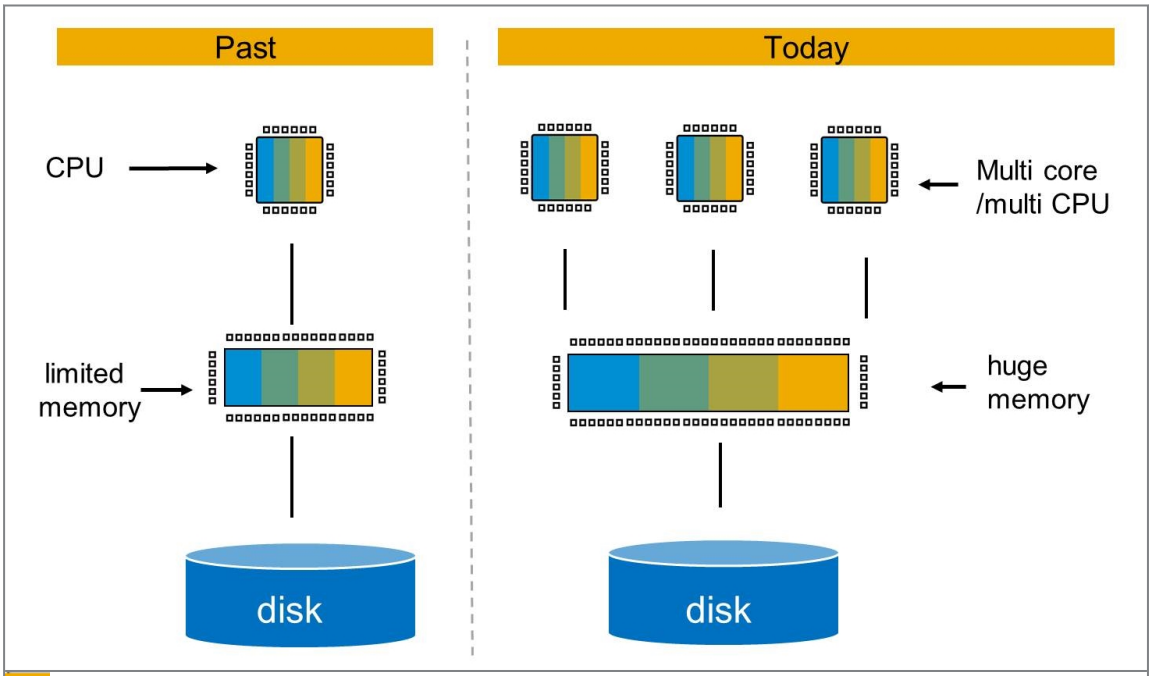


Figure 4: Huge Computing Power Now Available

With so much memory available, we can now store the entire database of even the largest organizations, completely in memory. This gives you instant access to all data, and eliminates wait times caused by data loading to memory from disk to memory. We can lose the mechanical spinning disk and the latency it brings, and rely on memory to provide all data instantly to the CPU. Even though Solid State Devices (SSD) storage is faster than disk it still can not compete with memory. So memory is no longer the bottleneck it once was. To address large amounts of memory, we also use 64-bit operating systems. Traditional 32-bit operating systems cannot address the large amounts of memory now available.

In addition to huge memory, CPU performance continues to improve at a phenomenal rate. We now have high-speed, multi-core CPUs that can take on complex tasks and break them up so then can be processed in parallel to provide incredible response times. This means that response times for even the most complex analytical tasks, such as predictive analysis, can be carried out in real time. So with huge memory and faster multi-core CPUs we have now have access to huge amounts of computing power.

A very recent development in computer memory was the introduction of a new type of memory. It is called **persistent memory** (PMEM).

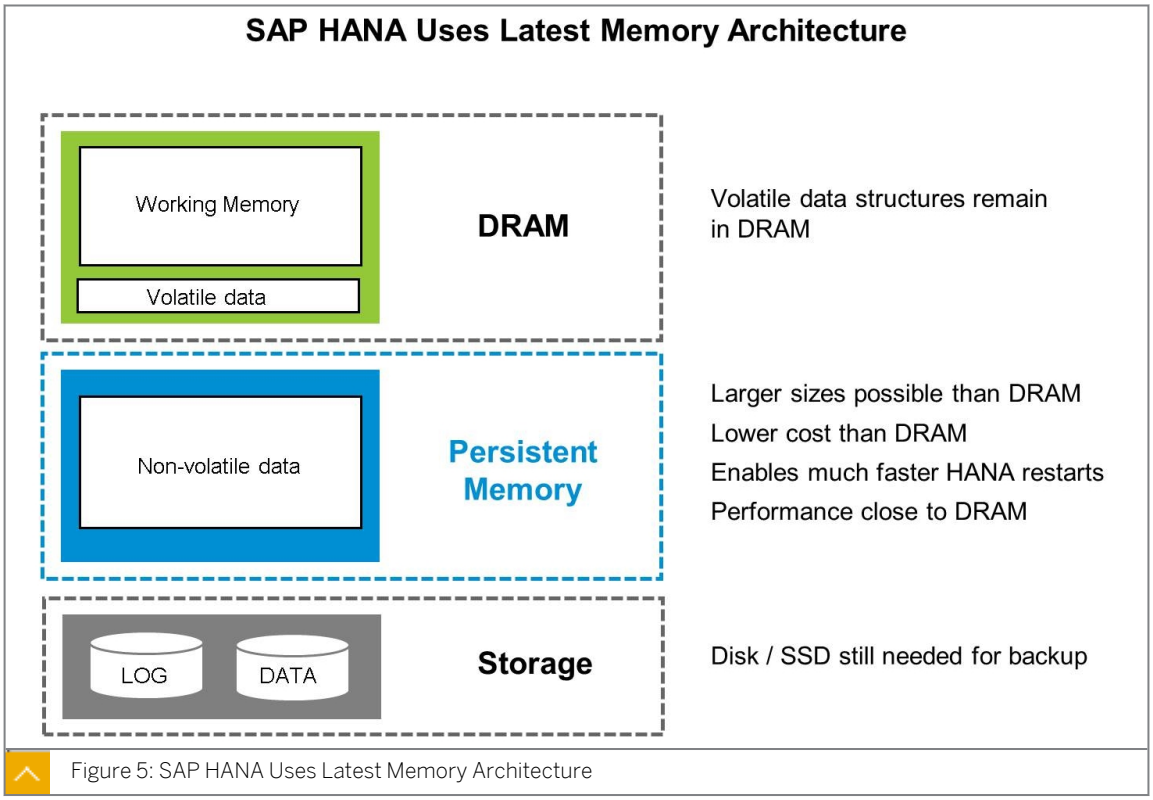


Figure 5: SAP HANA Uses Latest Memory Architecture

For decades, memory has been based on volatile, dynamic memory (DRAM). The upside to using DRAM for memory is the speed of data access for applications. But there are two downsides that come with using DRAM:

- Business Continuity — When power is lost due to unexpected outage or when SAP HANA needs to be restarted, perhaps after some maintenance, data is lost from the memory and needs to be reloaded from disk back to memory. With huge memory, reloading can take time and this means delays for customers and business users in resuming the use of their business applications.
- Cost— Even though DRAM prices are falling, it is still expensive to size hardware with very large DRAM memory.

PMEM is able to address both issues:

Firstly, PMEM is memory, just like DRAM, and is accessed by the applications in the same way. But unlike DRAM, PMEM is able to hold onto data even when there are power outages or the database needs to be restarted. This means when SAP HANA restarts the system is fully available significantly faster as the data structures do not need to be reloaded from disk to DRAM.

Secondly, PMEM is also cheaper than DRAM and an additional benefit is that PMEM comes in much larger sizes than DRAM. This means you can install huge memory at reduced cost compared to pure DRAM.

So should we completely replace DRAM with PMEM to save money and avoid long restarts? Quite simply: no.

DRAM is still the fastest type of memory available today and is ideal for storing the volatile data structures such as working data or calculations. This is why working memory is based on DRAM and not PMEM. PMEM is very fast but it cannot perform as well as DRAM. SAP recommends that customers implement DRAM alongside PMEM by selecting the optimal

ratio between DRAM and PMEM. SAP has defined recommendation for sizing DRAM and PMEM and you should take care to study these recommendations.

SAP HANA became the first database to support persistent memory at the application level. SAP worked closely with Intel to adapt SAP HANA so that it natively uses PMEM alongside DRAM. In other words, SAP HANA actively controls the movement of data between DRAM and PMEM to create a combined hardware (Intel) and a software (SAP HANA) solution.

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 handle huge workloads and data volumes. Once the limits of scale-up have been reached we can then look at scale-out landscapes where we deploy extra worker nodes to share the processing tasks.

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 available at that time. This meant they would not be able to fully exploit the power of the new hardware, with all the new developments previously mentioned. It would be like putting a racing driver with limited capabilities, into a high-tech Formula 1 race car where the potential of the car was never realized by the driver.

Put simply, the business software needed to catch up with advances in hardware technology. So, 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 and cache architectures. This enabled SAP to develop SAP HANA in such a way that it could extract every last drop of power from the hardware.

Total Database Transfer to Memory

In the past, databases were stored completely on disk and only the data requested by the applications would be moved to memory where it then passed to the CPU for processing. Due to its limited size, memory would soon become filled and so data in memory would need to be unloaded back to disk to make way for new data requests. A lot of disk swapping was normal but this was harmful to performance of the applications. Applications would be developed to try to reduce the swapping but this was never a proper fix to the problem, simply a workaround. With SAP HANA and the huge memory sizes available, you can now store the complete database in memory. This means that loading from disk to memory is not needed, all data is available instantly at all times to the application. The disk can remain unused.

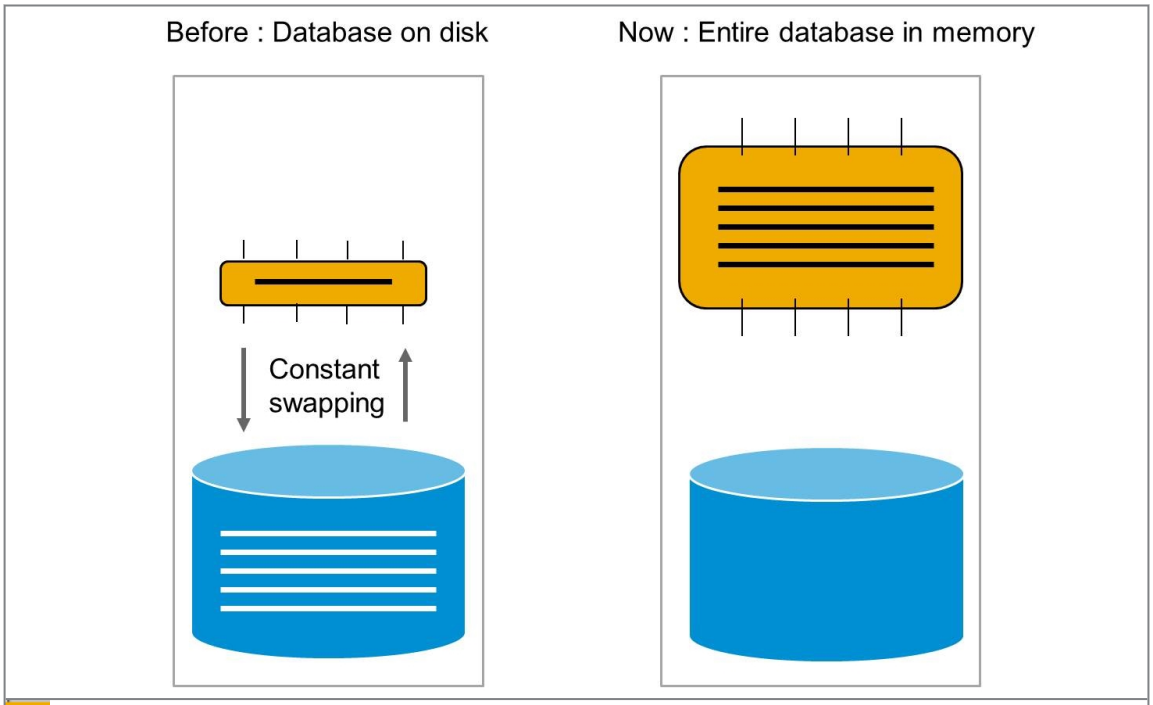


Figure 6: Moving the Database Completely to Memory

So how can we fit a complete database in memory? There are two key factors that work together to enable this:

- Huge amounts of memory are now available. We have progressed from gigabytes (GB) of memory to terabytes (TB) 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. In the region of 90%.

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 dataset to memory, only selected InfoCubes. So, you had to make a difficult choice as to which BW data you wanted to accelerate. SAP HANA accelerates all data because all data can fit it in memory.

So disk is not needed?

We know we can fit the entire database in memory, so does this mean we can eliminate disk? The answer is: no, we still need disk.

You need disk for the following reasons:

- Even though we can fit the entire database in memory, we usually don't want to do that.

  Data in memory is classified as hot, which means it is highly used, usually very recent data and needs to be closest to the CPU for optimum read performance. Infrequently used data can be classified as warm, which means that fast access is not so important. Warm data is stored on disk and loaded to memory as needed. Most organizations would not want all data in memory as they regard only a part of it to be hot. Memory costs are certainly falling but compared to disk, memory is still very expensive. This means that you should

deliberately size memory optimally to fit only the hot data and not worry about trying to fit the entire data of the organization in memory. When customers size their SAP HANA hardware they need to carefully calculate the memory size of the hot and warm parts of their data and choose a memory setup that provides great performance on the most important data and acceptable performance on the less important data.

- 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.

> Note:
> If data is stored in persistent memory (PMEM), then data is not lost during power outage; however, we still need disk as a reliable backup in case of hardware changes that mean even PMEM emptied.

Both of these requirements are 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 and is a requirement that the hardware used to support SAP HANA includes disk.

Simplified Data Models and Applications

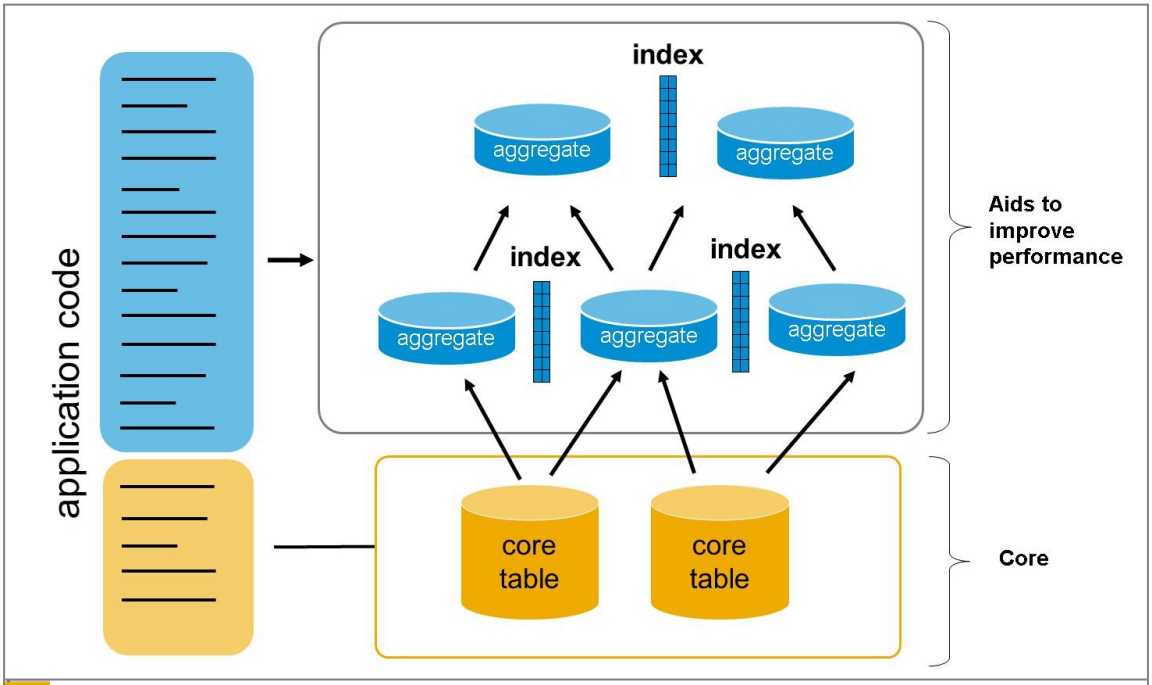


Figure 7: Traditional Approach to Data Model and Application Code

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 the modern fast moving business environment.

Eliminate Complexity of Applications with SAP HANA

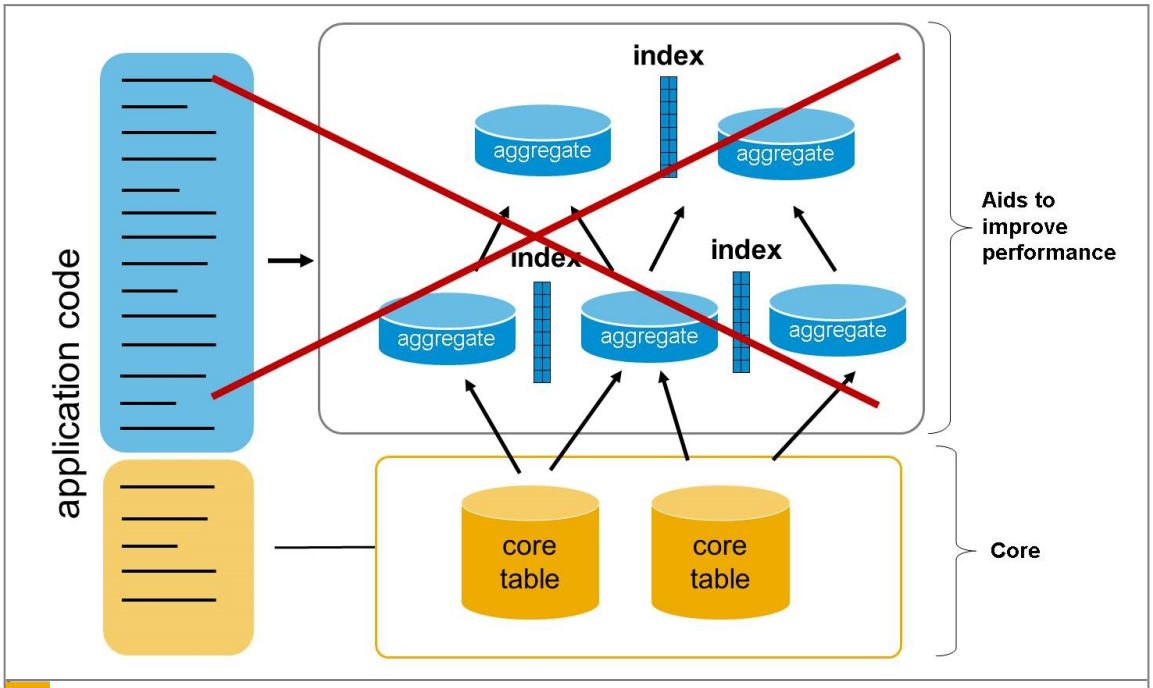


Figure 8: Eliminate Complexity of Applications with SAP HANA

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

SAP HANA organizes data using column stores, which means that indexes are 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.

It is now much easier to enhance the applications and integrate additional functions.

Movement of Processing to the Data

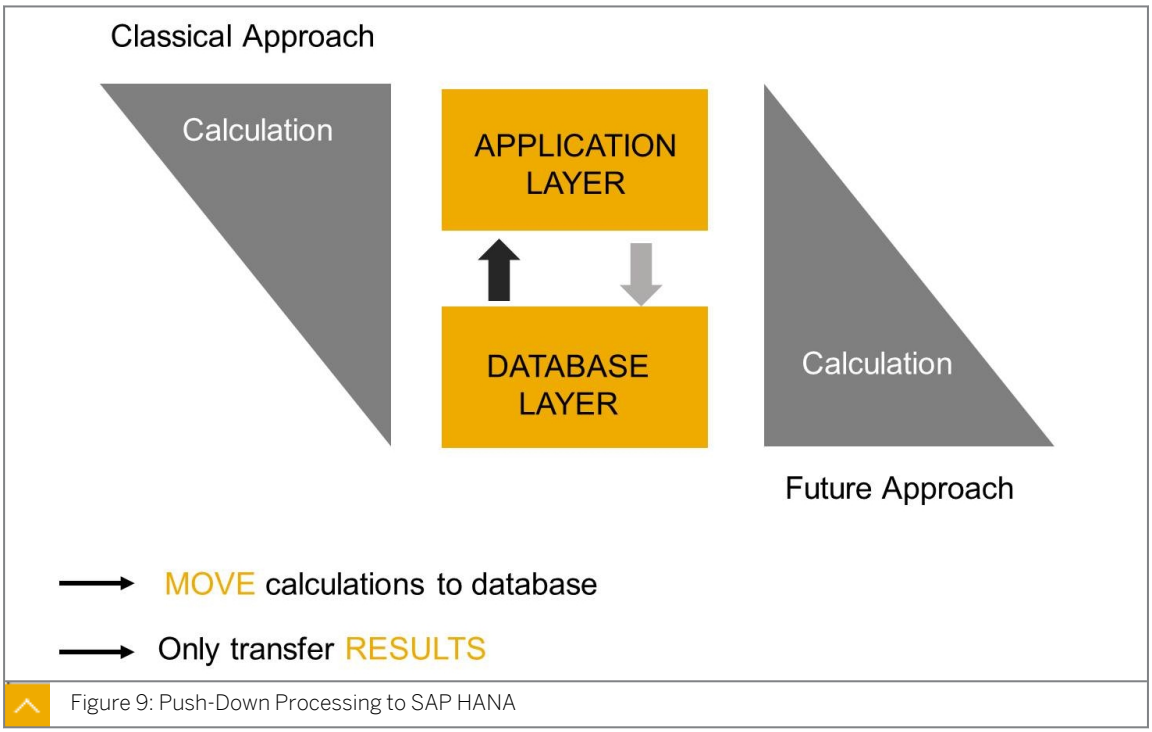


Figure 9: Push-Down Processing to SAP HANA

In the past, the role of the database layer was to listen out for data-related 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 requested only 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 work on simple data-related tasks, it wants to handle the complex tasks too. These tasks can include the following:

- Aggregate and disaggregate data
- Filter, sort and rank data
- Calculate values from data
- Generate forecasts
- Convert data (for example, currency, unit of measure)
- Cleanse data

- Re-structure data

Push-Down Process

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 less data passes between the database and application layer.

For example, if the application sent the instruction to SAP HANA: **`Please summarize the last 5 years' sales line items of yellow widgets by region with a breakdown by year, and calculate the net value after discounts applied`**, instead of sending millions of raw rows of data from the database to the application layer, SAP HANA processes the data request and sends back only the results to the application layer. This means a huge reduction in data volume being passed from the database to the application server. As well as this benefit, all data processing is performed in memory by SAP HANA, so performance is excellent.

Moving the data processing tasks from the application layer to the database layer is called push-down. Applying the push-down approach means that application developers need to re-think the way they code. In the past, all coding was done in the application layer, but now with SAP HANA, large parts of the coding can be developed directly in the database.

Note:
For simple applications that include limited business and user interface logic, SAP HANA can take care of all processing and a separate application server is not needed. This is because SAP HANA has a built-in application server called *SAP HANA XS Advanced (XSA)*. We will cover this topic later in the course.

However, the application layer is still needed for complex, enterprise applications such as SAP S/4HANA and Business Warehouse (BW). The application layer is needed to handle the complex business logic that is usually programmed in a dedicated business programming language. For SAP applications, the programming language is ABAP.

One Data Set for All Applications

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 by all applications.

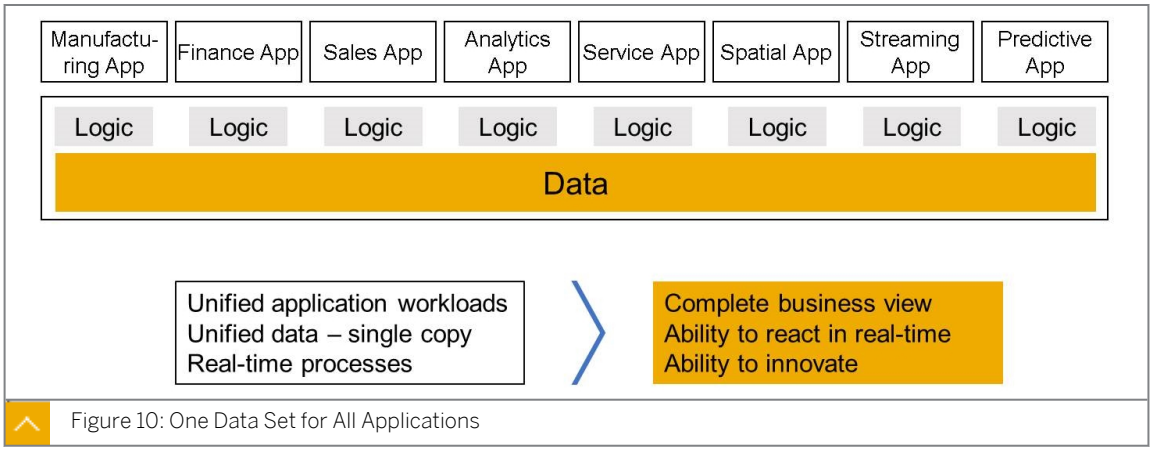


Figure 10: One Data Set for All Applications

Traditionally, applications were powered by their own dedicated platforms. In order to share data between applications, data would need to be transferred, often through complex middleware and software to clean and restructure data. This lead to latency, silos of data and

of course, complexity. A better approach is to have all applications powered by one high performance platform. This means a common architecture with only one data management solution where all data is available to all applications in real time. Then there is no more unnecessary data movement or management of multiple data stores and no more complex platform-to-platform architecture with high maintenance interfaces.

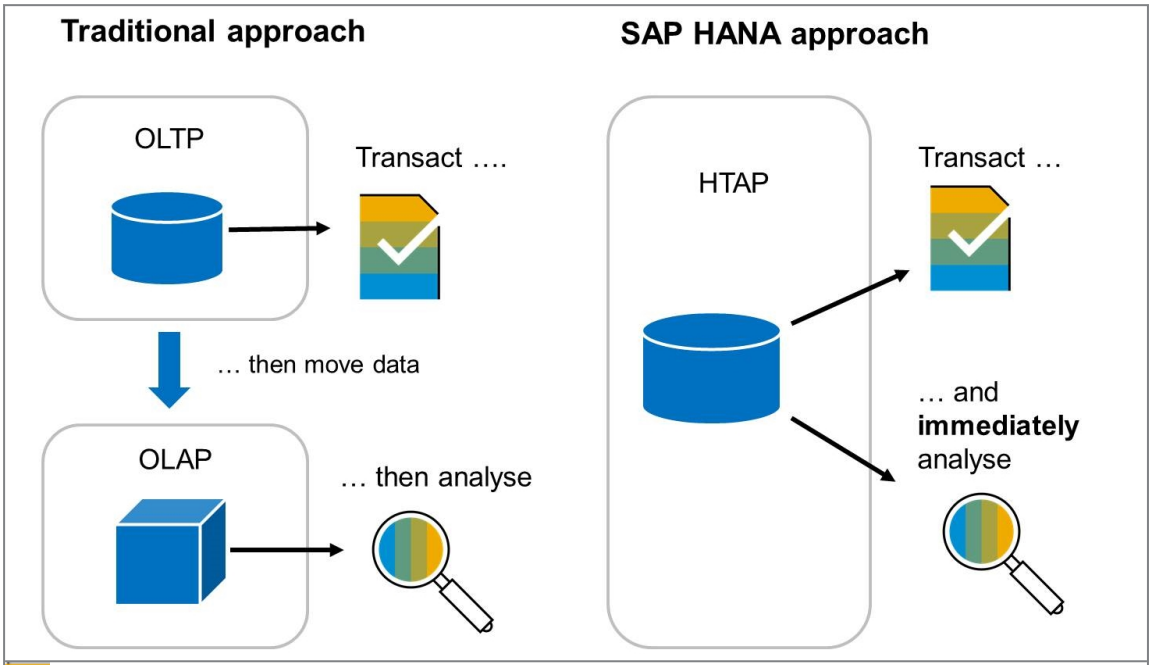


Figure 11: Bringing Analytics and Transaction Processing Back Together

In many business landscapes today, transactions are managed in systems where both the hardware architecture, database design and the data models are built around fast read/write processing at the record level. Analysis systems take on a different design approach. The hardware, database, and data models are built around batch loading, aggregated storage, and a focus on read-intensive queries and caching. That is why online transaction systems (OLTP) and online analytical processing systems (OLAP) are often separated and linked through interfaces through which data is lifted and shifted periodically. This approach means delays in analysis on the transactional data. It is not unusual to have to wait the next day before analysis can begin on the transactional data.

SAP HANA is able to bring transactional and analysis requirements into one platform. The acronym for this type of consolidated system is Hybrid Transaction/Analytical Processing (HTAP).

The database, hardware, and data model of SAP HANA are built to handle combined transactional and analysis processing. No movement of data is necessary and transactional and analytical users work from the same, single copy of the data. This means we have live data available to all applications in real time. This reduces the complexity by removing the need to move data using separate software and interfaces. It also means that new innovative applications can be built that combine transactions and analytics such as those found in SAP S/4HANA.

For example: An employee vacation booking system that dynamically calculates real-time forecasted peak workload times so the employee can be sure to avoid booking vacation at a time that might impact the business. In the past, two separate applications would need to have been built on separate platforms: One for analysis and another to execute the transaction. This new approach has revolutionized the way applications are built and the line between analysis and transaction application has become blurred.

Access to Any Type of Data Regardless of Location

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 new data types:

- Text Data

  This is unstructured data from social media posts, help desk tickets, news streams, system 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.

- Series Data

  This is data that is collected in a steady time continuum, such as smart meter readings captured every 10 minutes.

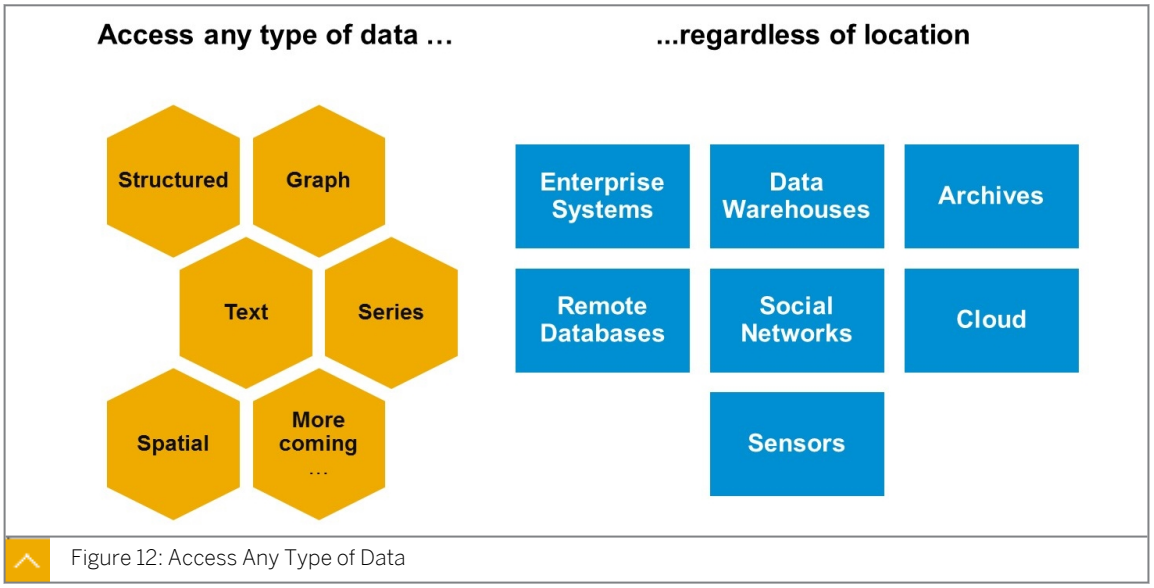


Figure 12: Access Any Type of Data

SAP HANA can process all types of data and also combine them in new and innovative applications.

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

- Enterprise Systems

  SAP 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.

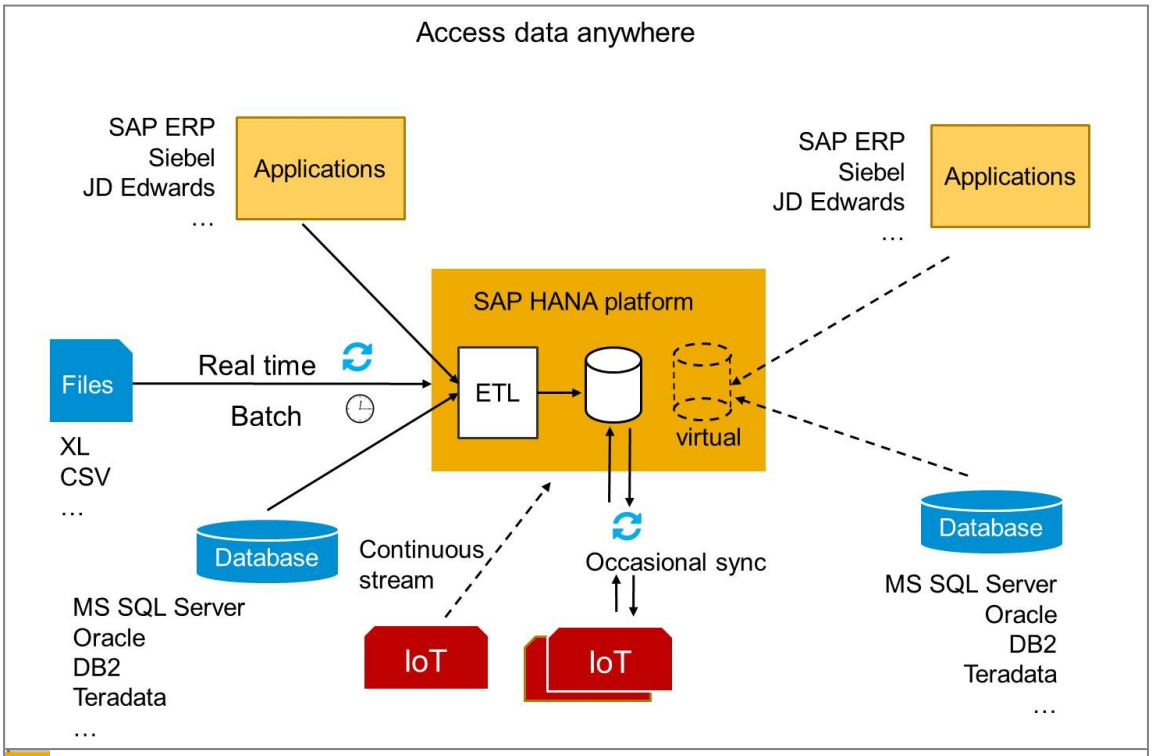


Figure 13: Access Data Anywhere at Any Time and Anywhere

SAP HANA can consume data for processing from any location. SAP HANA is able to store huge amounts of data locally in its database. Data can be extracted from any source using built in ETL tools that can clean and integrated data from multiple sources and loaded to SAP HANA for storage. Data loading schedules can be set up to automate the loading tasks.

Data can continually be streamed to SAP HANA or extracted and loaded to SAP HANA in real time or batch.

SAP HANA can also access remote data without loading anything to its database. This means a live view of remote data is always possible. To the application running on SAP HANA, it treats even the remote accessed data as if it was stored locally.

Platform Services to Power any Type of Application

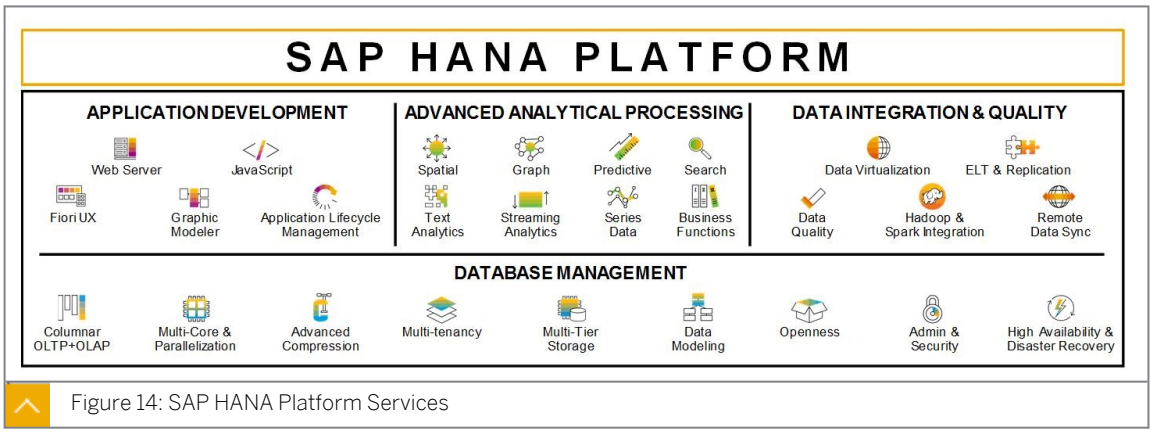


Figure 14: SAP HANA Platform Services

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 SAP HANA platform includes all components to power any application. These components include database, data processing, application development design time and run times, lifecycle management, and data integration. They can be used in both SAP and non-SAP applications.

Key Capabilities of SAP HANA

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

- Application Development

  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 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 runtime is there.

- Advanced Analytical Processing

  SAP HANA can handle many new types of data. This includes text, spatial, graphic, and more. However, it is not enough to 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.

- Data Integration

  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 Management

  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.

SAP HANA Powers the Intelligent Enterprise

Organizations demand more from their enterprise suites. In the past, the key role of the enterprise suites such as SAP R/3 and later SAP Business Suite, was to provide a system of record. But today they are expected to work much harder and to provide intelligence to the business. This means enterprise suites should provide in-line suggestions to business users during transactions, make recommendations that reduce costs, improve profitability and productivity, guide the optimization of the business resources, identify and avoid risk and much more.

Intelligent enterprises use new data types that can be processed in real time from any sources to feed in to the advanced machine learning algorithms that build the intelligence of the enterprise through machine learning.

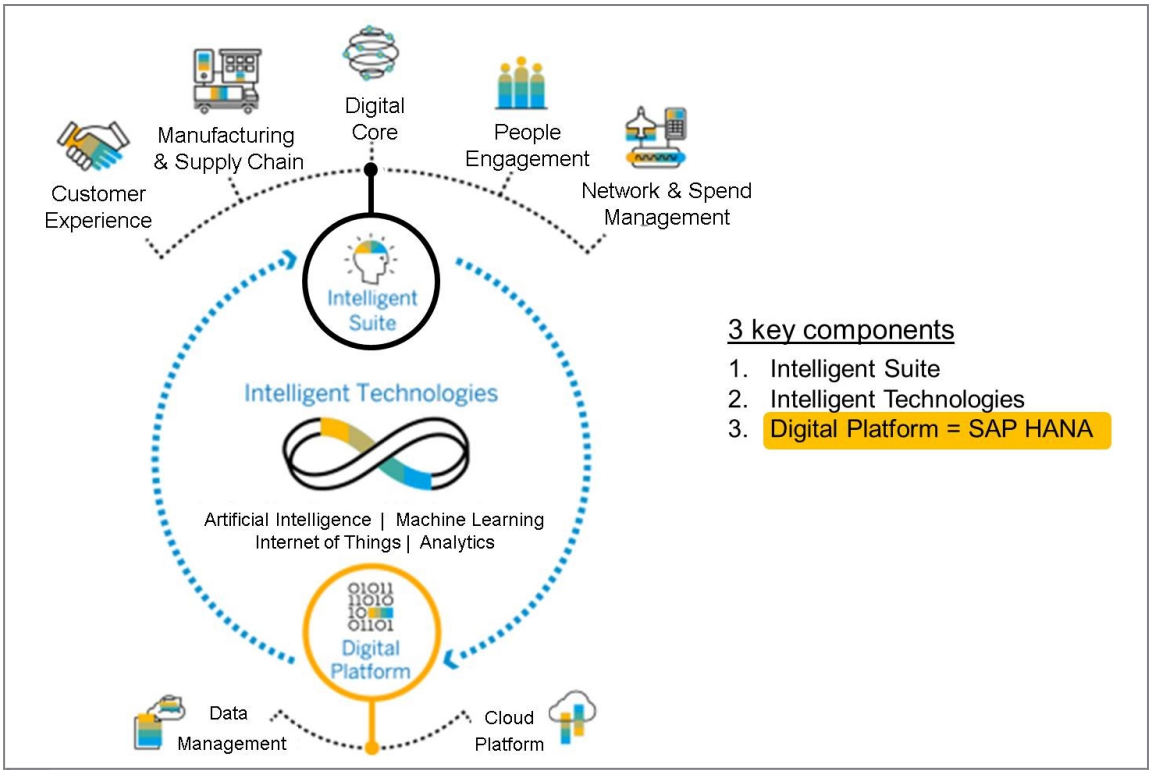


Figure 15: Intelligent Enterprise

But to do all this, the intelligent enterprise needs to run on a powerful digital platform that can provide all advanced services needed. For the SAP Intelligent Enterprise, the digital platform is SAP HANA.

SAP HANA Use Cases

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

SAP HANA can be used in a large number of different IT scenarios. Let's take a look at some examples of where SAP HANA can be used.

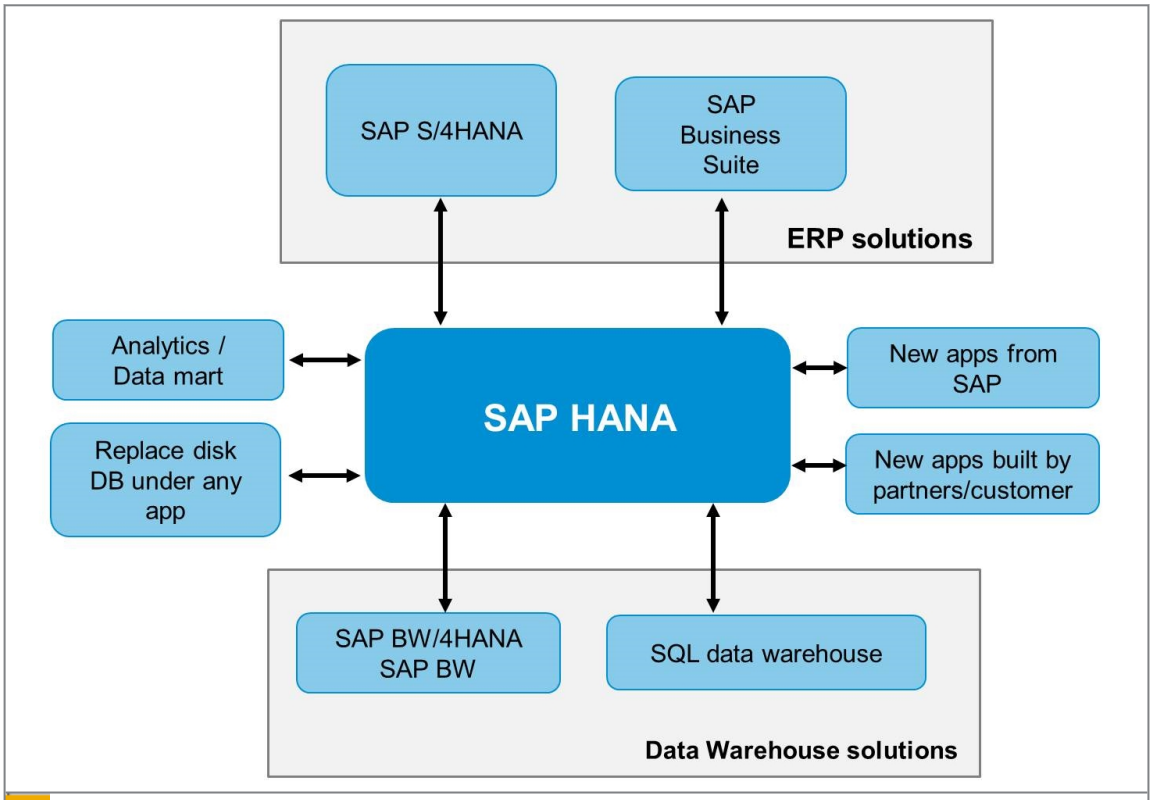


Figure 16: Use Cases for SAP HANA

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

- To power enterprise suites such as SAP S/4HANA and SAP Business Suite.
- To replace any relational database.
- To power off-the-shelf enterprise data warehouses such as SAP BW/4HANA or SAP BW, or to build bespoke SQL data warehouses.
- To power partners' and customers' new applications.
- To build local data marts from scratch that can provide real time analytics.
- To power innovative line-of-business SAP packaged applications that are built to run only on SAP HANA, such as Sports Analytics or Smart Meter Analytics.

We cover each of these use cases in more detail in a later lesson.

LESSON SUMMARY

You should now be able to:

- Describe how SAP HANA powers a digital platform

Unit 1
Lesson 3

Key Technologies of SAP HANA

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

- Describe the key technologies of SAP HANA

Row and Column Tables

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 based on row storage, as this is regarded as the optimal storage design for an on-line transactional processing (OLTP) application. An OLTP application requires fast, record-level updates where all columns in the record are usually needed for processing.

SAP HANA fully supports OLTP applications using row storage but also supports columns based storage which is the optimal storage design for on-line analytical processing (OLAP) applications. OLAP applications typically work with high-volume tables that need to be quickly aggregated by ad-hoc queries.

The database of SAP HANA supports both row tables and column tables in the same database. Modern applications, such as SAP S/4HANA, combine transactions and analytics so SAP HANA, with its row and columns storage, is the ideal database on which to build such applications.

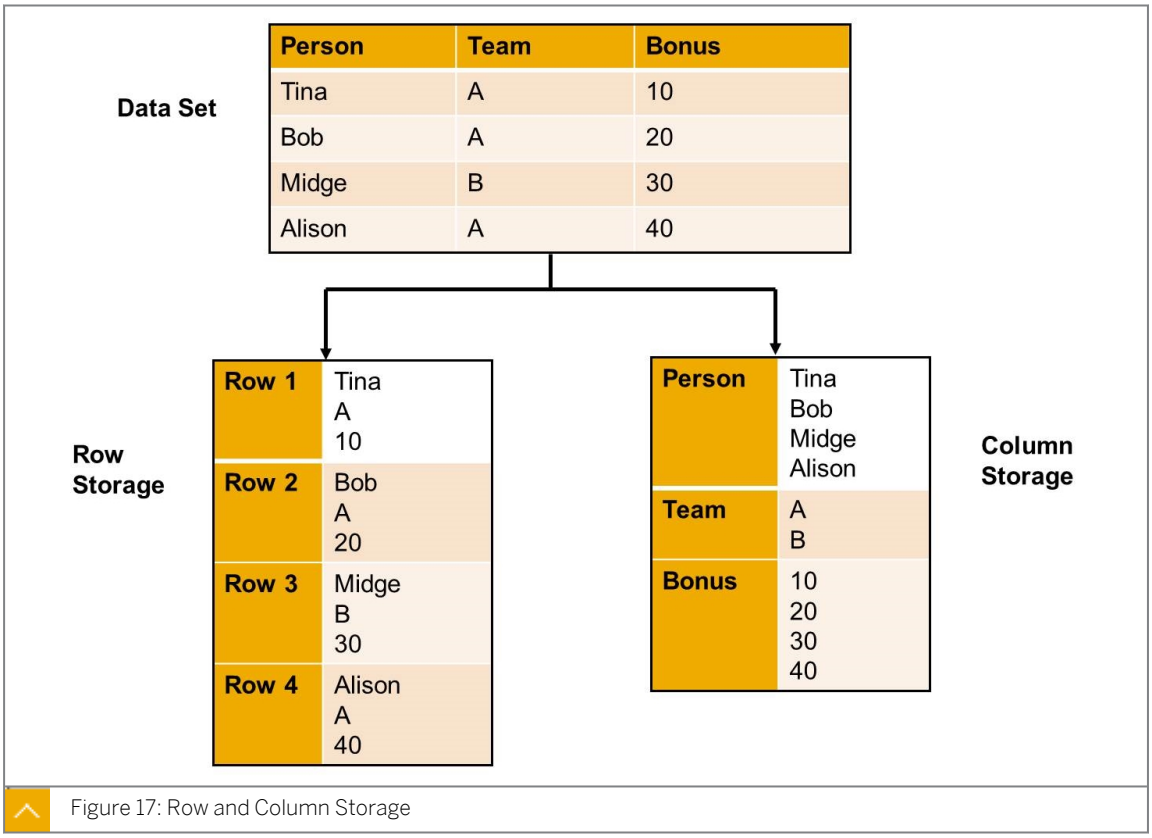


Figure 17: Row and Column Storage

The figure, *Row and Column Storage*, illustrates how row and column tables store the data.

Column tables are efficient for analytical applications where requests for selections 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 a few columns are required from the table and also only a limited number of entries from the columns are needed. With column tables, only the required columns are loaded to memory, so you avoid filling 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 unwanted values.

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

It is easy to alter column tables, for example, by adding extra column or removing columns, without dropping and reloading data.

Column tables are optimized for parallel processing, as each CPU core is able to work on a separate column.

The downside to column tables is the cost of reconstructing complete records from the separately stored columns. Reconstruction typically occurs in transactional applications that requires the complete record for updating, copying, or deletion. Although it is possible to build transactional application on column tables, you might see better performance if you were to use row-based tables where all the columns in a record are always held together and can be read in one step.

But more often these days applications combine transactional and analytical elements. In this case, you must decide which is the best storage method to use. You cannot have a table that is both row-based and column-based.

Note:
Converting a column table to row and vice versa is a simple and quick operation at the SQL level (ALTER TABLE) and you do not lose the data when doing this.

Data Footprint Reduction

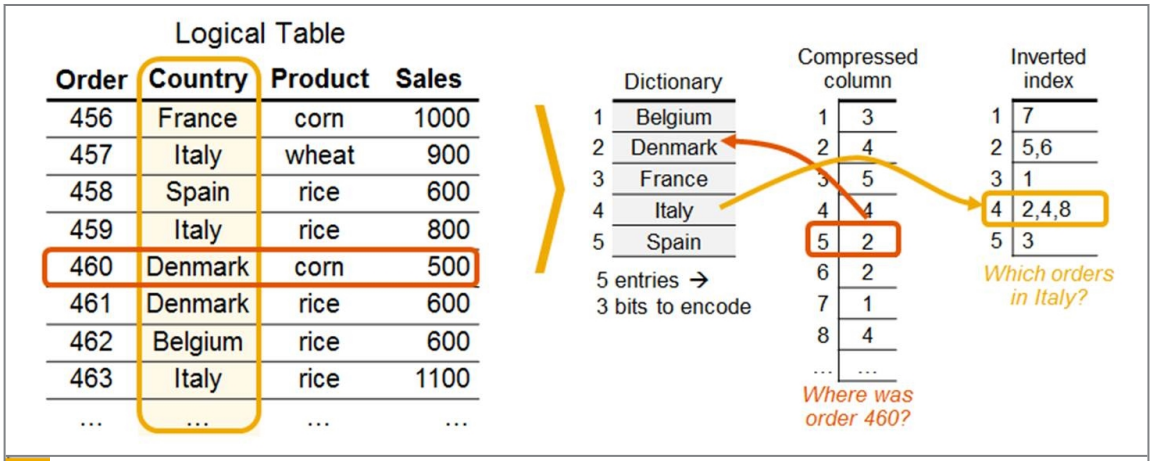


Figure 18: Compression

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.

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.

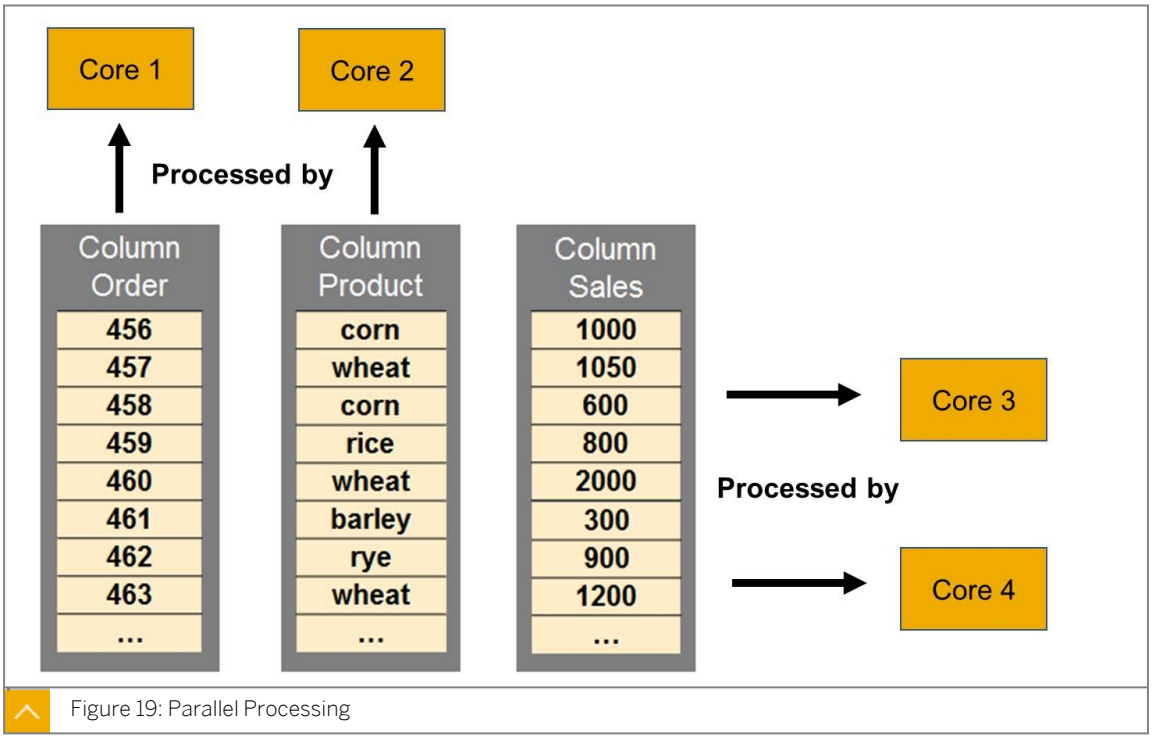


Figure 19: Parallel Processing

SAP HANA automatically spreads workloads 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.

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.

Active/Active-Read Enabled Mode

It is possible to replicate the SAP HANA database log to a secondary instance of HANA and then to asynchronously replay the database log in the secondary system. This keeps the two HANA databases 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.

Built on top of this technology, SAP introduced active/active-read enabled mode.

Active/Active-Read Enabled Mode

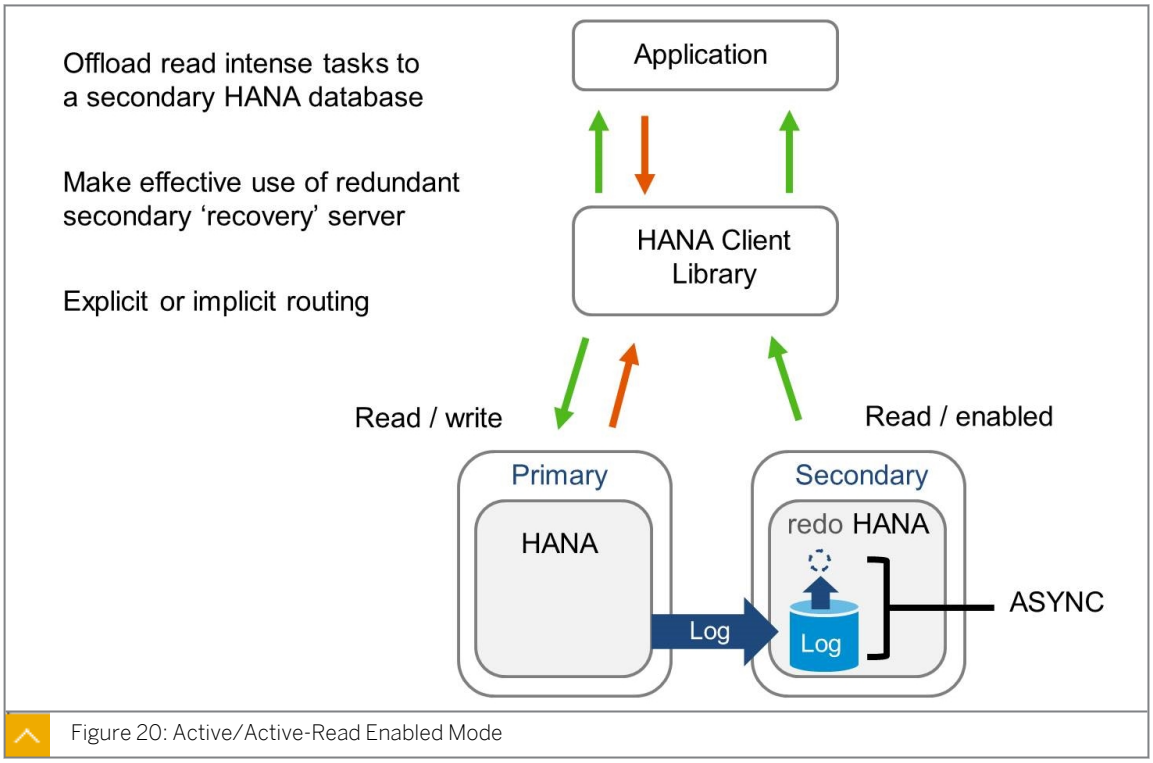


Figure 20: Active/Active-Read Enabled Mode

With active/active-read enabled mode, you can redirect applications to use the fully-synced, secondary SAP HANA database specifically for their read-intensive operations, whilst continuing to use the primary system for write-enabled operations. This improves the overall performance of the SAP HANA database by providing a better balance of workloads.

Applications can either be hard-coded or use implicit hint-based routing to determine where the read should take place.

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

Note:
As of SAP HANA 2.0 SPS04, it is possible to implement multiple secondary systems with active/active-read mode enabled so that you can distribute the read-intensive workload evenly across your landscape to improve overall performance.

Maintaining Good Read Performance with Frequent Database Updates

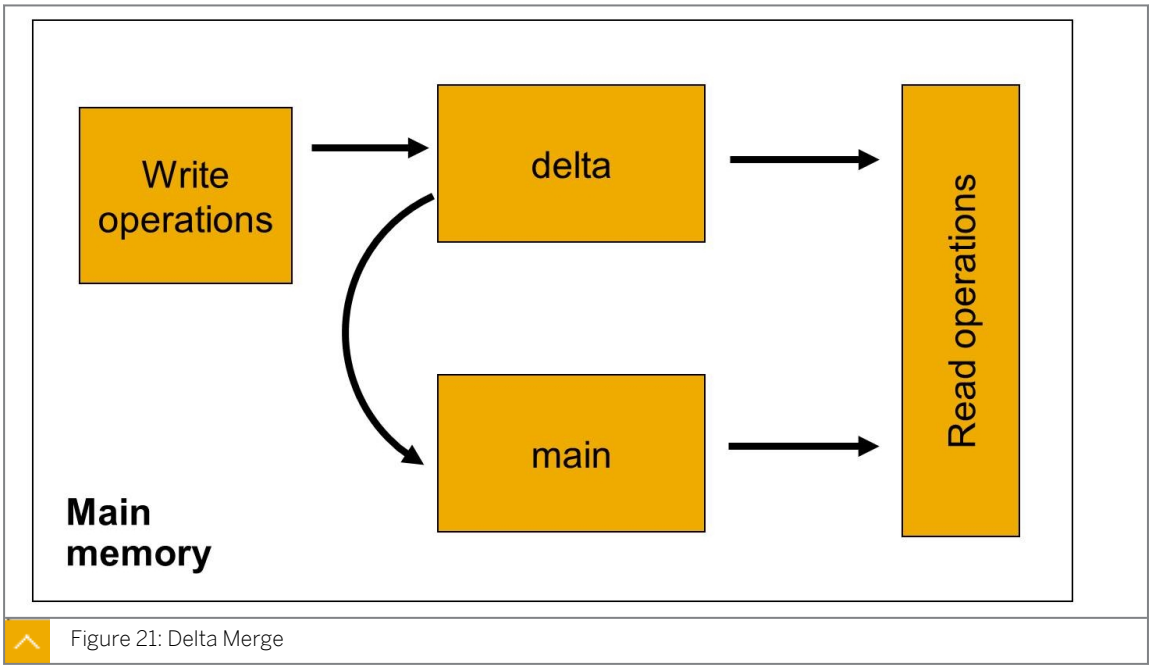


Figure 21: Delta Merge

One of the benefits of column tables versus row tables is that column tables are automatically compressed in order to reduce the data footprint and improve read performance. However, updating and inserting data into a compressed column table is a costly activity. This is because each column has to be first uncompressed, the new records inserted and then re-compressed again. Thus, the whole table is reorganized each time. This constant re-organization would seriously harm performance.

To deal with tables that are column-based but need to support updates and insertions, every column table has a *main store* which holds the compressed, read-optimized, sorted data and also a *delta store* which holds write-optimized, non-sorted data. New records always arrive in the *delta store*. A periodic, automated database activity merges the *delta store* into the *main store*. This activity is called a **delta merge**.

So that even the newly arrived data is available immediately, queries always run against both *main store* and *delta store* simultaneously. The *main store* is the largest one, but because its data is compressed and sorted, it is also the fastest one. *Delta store* is very fast for insert as the records are not compressed, but it is much slower for queries, and therefore is kept relatively small by running the delta merge frequently. The delta merge is a background tasks and does not interrupt the use of an application.

The delta merge can be triggered based on sizing conditions that can be set. For example, you could define a condition that checks if the delta store size is greater than 5% of the main store size. If so, the delta merge is triggered. Also, some applications, such as SAP BW, can trigger a delta merge on demand, for example after a data load has completed overnight.

Delta merge can also be triggered by an application. Staying on top of the delta merge is critical to maintaining good performance of SAP HANA as the *delta store* can quickly become very large as the data is not compressed. The administrator is responsible for this and has tools to monitor the *main store* and *delta store.*

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

Multi-Tenancy

SAP HANA can run multiple, isolated applications within the same system. This is possible because the database of SAP HANA supports Multi-Tenancy.

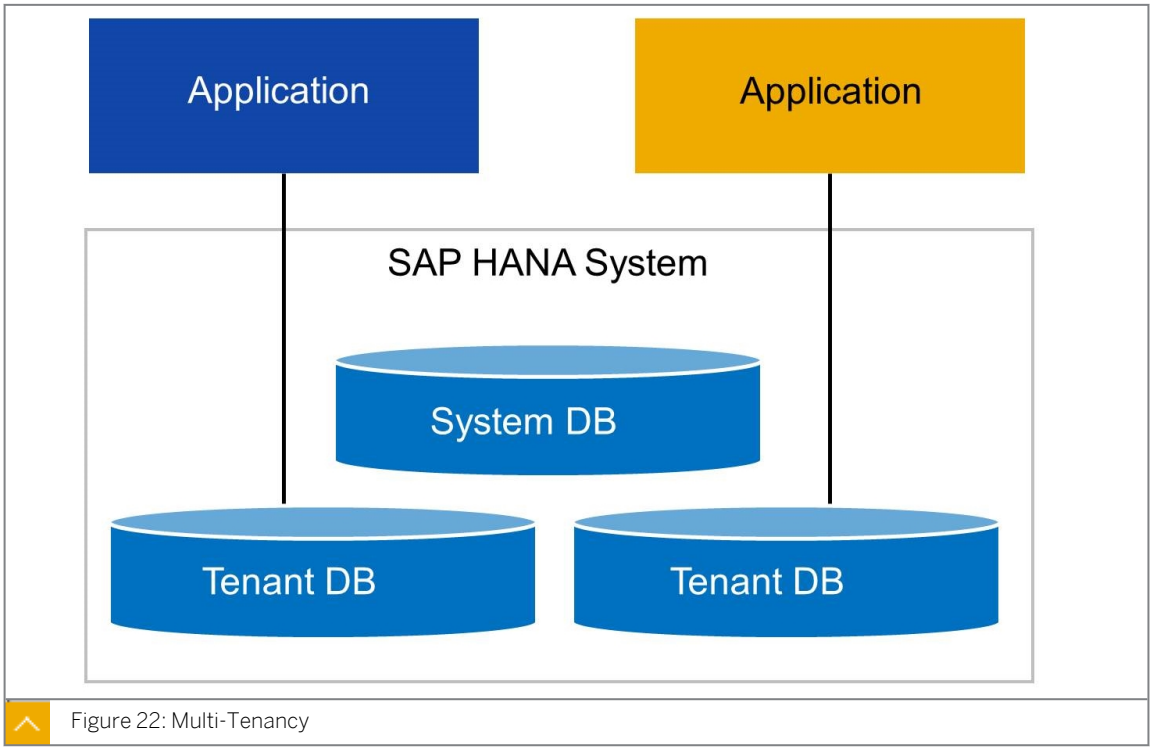


Figure 22: Multi-Tenancy

With Multi-Tenancy, there is a strong separation of business data and users profiles who must be kept apart. Each tenant has its own isolated database. Business users would not be aware that they are sharing a system with others running completely different applications.

In a Multi-Tenancy setup there is always a *System* database which is used to manage the system-wide settings and cross-tenant operations, such as backups. You can then set up multiple tenant databases to support each application.

The key benefit of a Multi-Tenancy platform is that we can host multiple applications on one single SAP HANA infrastructure and share common resources to simplify and reduce costs. But each application is completely isolated from a data perspective but also from a resource perspective. This means each application is assigned its own chunk of SAP HANA processing power and this does not interfere with other applications.

Note:
There is no need for SAP to provide Multi-Tenancy for the SAP HANA cloud service because the same outcome can be achieved by simply deploying new instances of SAP HANA in the cloud. Each instance is completely isolated and can be sized as required.

Catalog versus Container Approach for Database Object Management

When SAP HANA 1.0 was released, all references from the data models and application code to the database objects (for example, tables and functions) needed to include the database schema name. So if a table was needed in an SQL statement, it would look like this:

```
SELECT * FROM "MY_SCHEMA.SALES"
```

This approach meant that if a table was moved to a different schema the code would fail. Hard-coding schemas to code and data models is also a problem when it comes to supporting multiple deployments of the same application. How could you have different versions of a table using this catalog/schema approach? You would have to manually create separate schemas in the database with different versions of the tables in each schema, and then have two versions of the application code, each pointing to the different schemas. This is called the **catalog** approach and it still supported in SAP HANA 2.0 using the SAP HANA 1.0 technology.

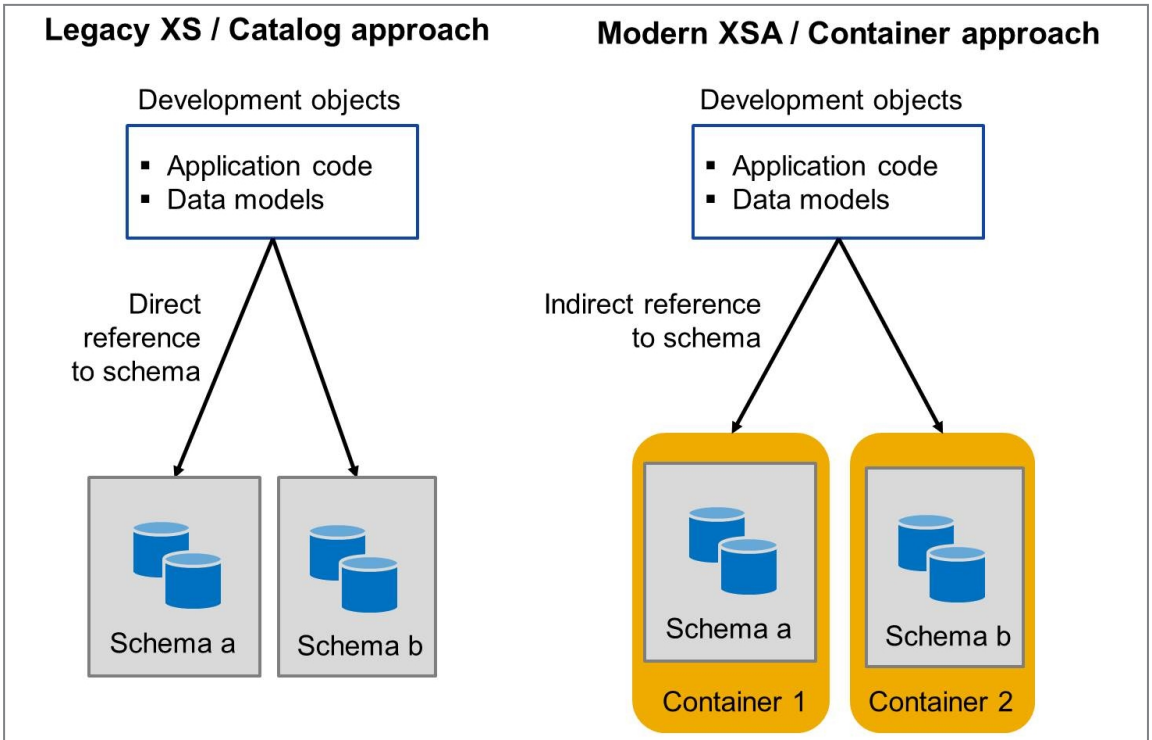


Figure 23: Catalog vs Container Approach

For SAP HANA 2.0, a new approach was introduced for deployment of an application. Fixed database schema references were no longer needed. Instead of referring directly to database schemas, we now write code and develop data models schema-less. What this means is that a developer never needs to be concerned about the physical schema name to which their database objects are deployed. All database objects that belong to an application are deployed to a **container**. A container is a run-time object that is automatically generated the first time you build your database run-time. Technically, a container is a logical layer that sits above the physical schema. The physical schema is automatically generated when the container is generated but it is hidden from the developer as they do not need to be concerned with it. So now you can deploy an application multiple times and each deployment can generate a new container (and therefore individuals schemas) to ensure isolation of database objects. This means database objects can exist at different versions.

So, using containers, the SQL code would look like this:

```
SELECT * FROM SALES
```

Notice there is no reference to a fixed schema any longer? The developer does not need to get tied up in schemas, they simply code schema-less.

As we move the application code to a new version and a new container is generated during the build of the application, the code follows the location of the new table. The container is a key component in the new **SAP HANA Deployment Infrastructure (HDI)** technology of SAP HANA 2.0.

The catalog approach is tied to XS developments and is supported by SAP HANA Studio. The container approach is tied to modern XSA developments and can only be used with Web IDE for SAP HANA, and not SAP HANA Studio.

Note:
The HDI technology was actually introduced with SAP HANA 1.0 SPS11 but many improvements were made with SAP HANA 2.0.

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 that are comprised of web-based or mobile user interfaces (for example, HTML), business logic (for example, JAVA), and extensive SAP HANA data models (for example, calculation views). SAP Web IDE for SAP HANA works in conjunction with the SAP HANA deployment infrastructure (HDI) and the XS Advanced runtime platform. It does not work with XS Classic development and so it is associated with SAP HANA 2.0 developments.

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 similar client tool, but it is used for the development of SAP 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 and online references (especially from unofficial SAP sources) as these products are easily confused.

SAP Web IDE for SAP HANA provided the following features:

- Integrated Workspace and Project Management
  - 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 includes 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*, displays.

Launching Web IDE for SAP HANA

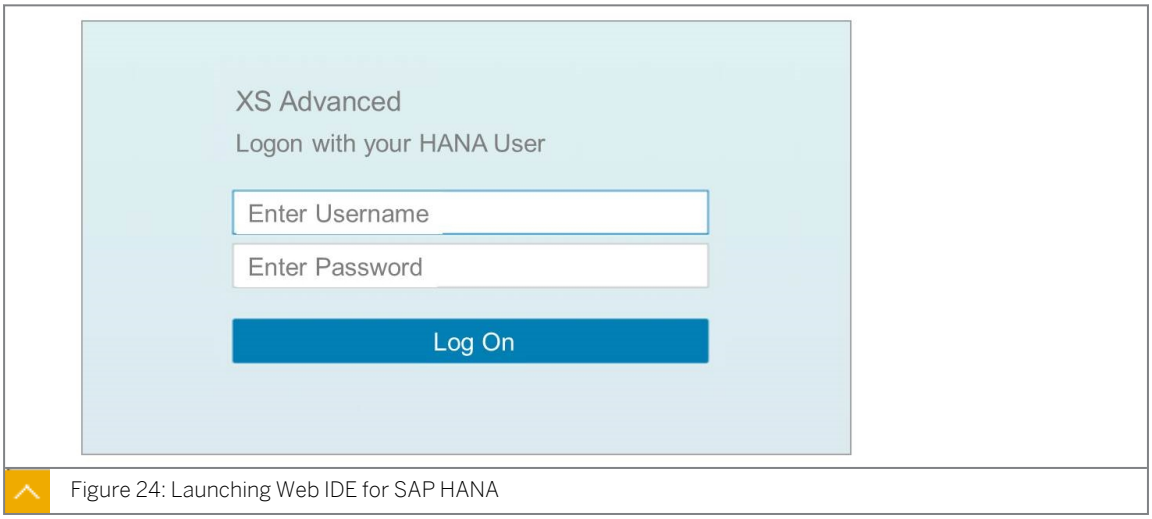


Figure 24: Launching Web IDE for SAP HANA

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

Switching Views in Web IDE

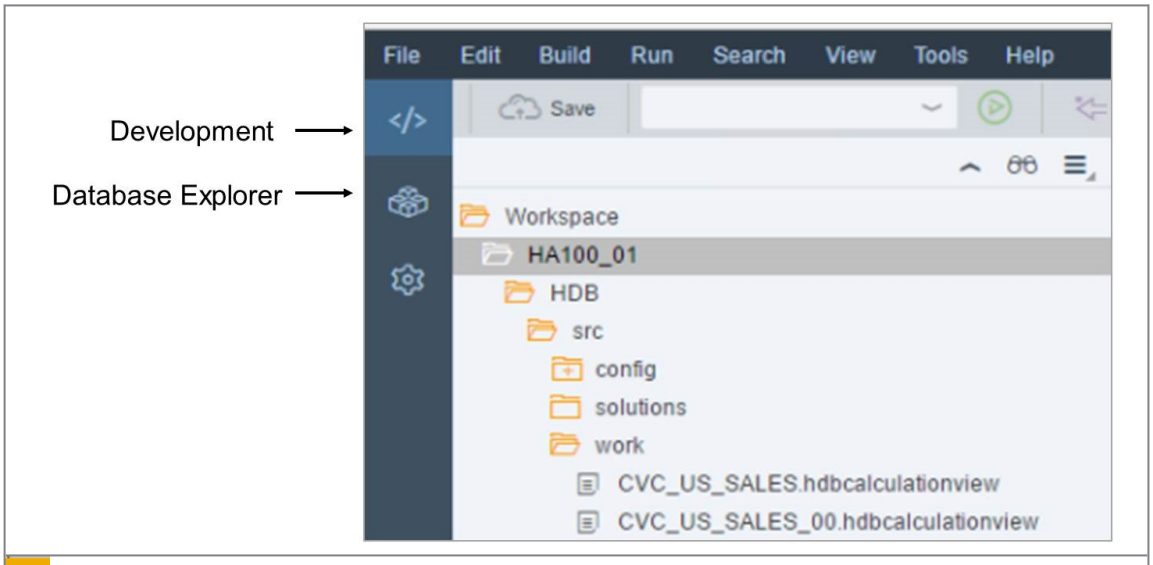


Figure 25: Switching Views in Web IDE

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

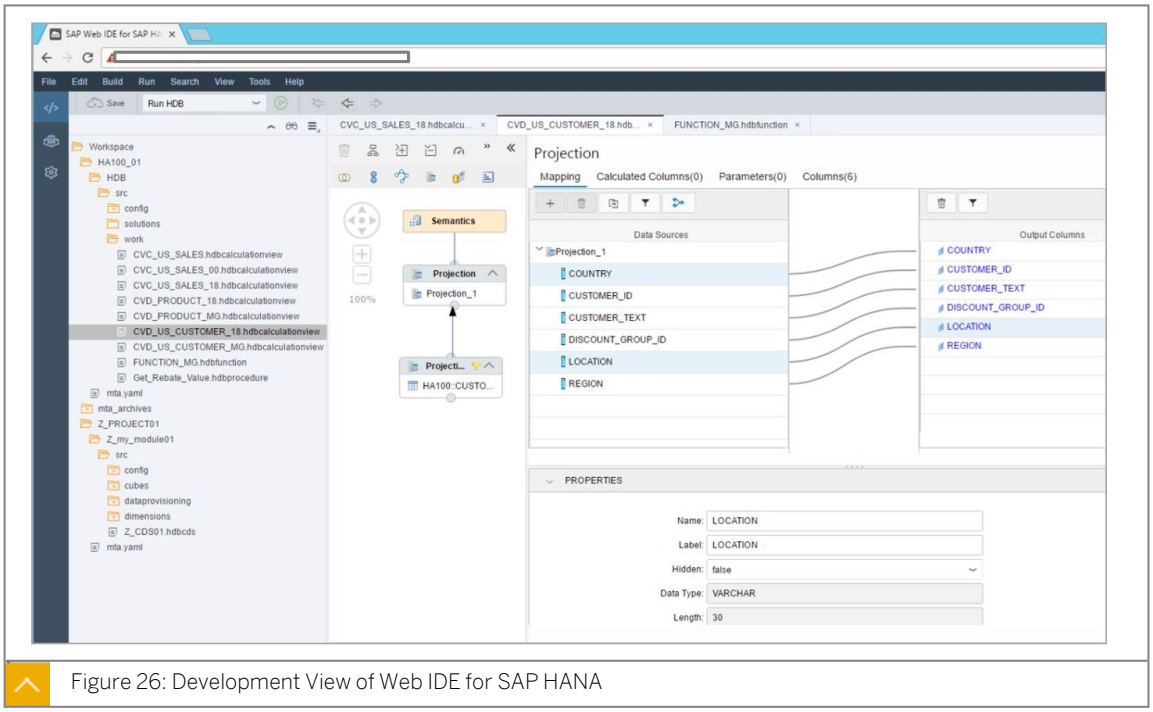


Figure 26: Development View of Web IDE for SAP HANA

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, debuggers, and syntax error highlighting.

Creating a New Development Module

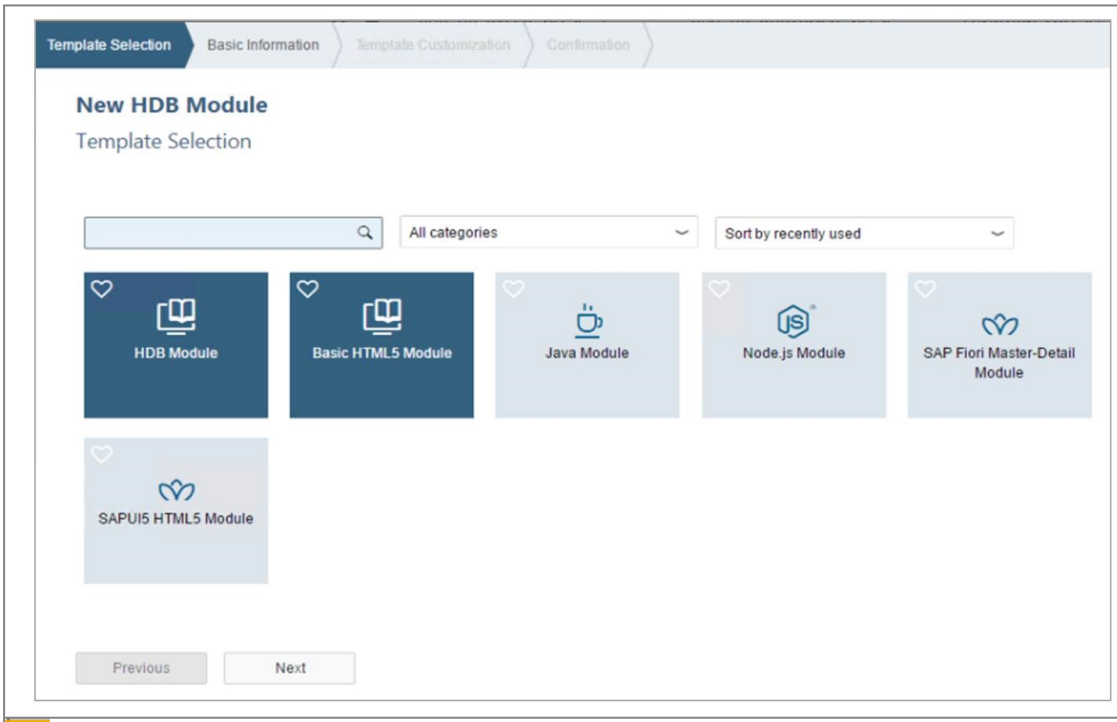


Figure 27: Creating a New Development Module

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

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

To create the various development artifacts and source files, you right-click a folder and choose the type of file you want to create from 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.

Exploring the Database with Web IDE

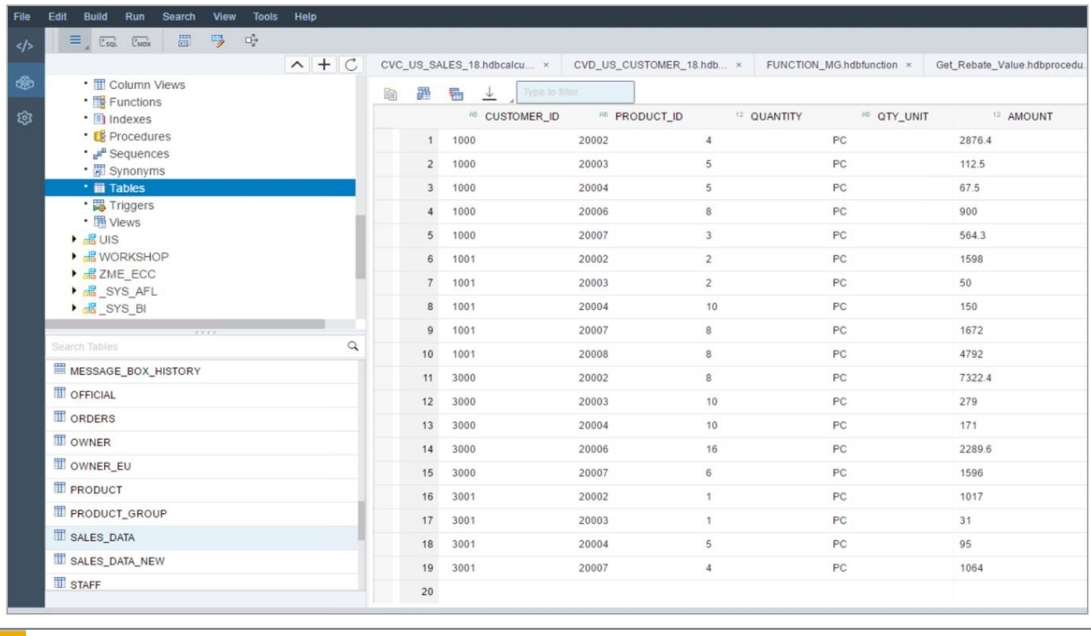


Figure 28: Exploring the Database with Web IDE

The *Database Explorer* view is used by anyone who needs to view the runtime database objects such as tables, views, and functions. There are tools to view the metadata and also the data of tables.

The *Database Explorer* also includes an SQL and MDX console so you can enter and execute statements directly against the database.

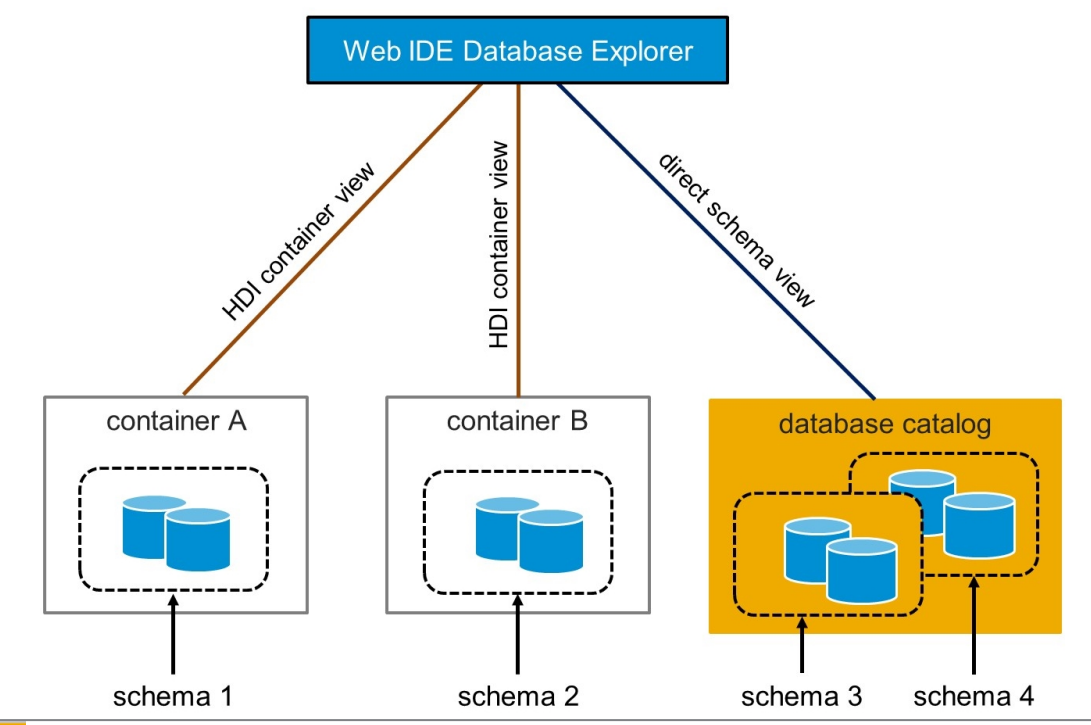


Figure 29: The HDI Container or Schema View

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 are 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.

The classic schema-based approach is the traditional way we view the list of database schemas and then expand these to reveal the database objects they contain. This is the only way you can view the database using SAP HANA Studio.

But let's briefly describe an **HDI container** as this is the newer approach to accessing the database, and whilst it may seem a little complex at first, you soon discover that this approach is simple and makes much better sense for flexible development and deployment.

An 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 database schema is automatically generated and bound this container. So developers simply refer to the container and they must never specify the physical schema name. (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 are 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 is used for the newer applications database objects. The older application continues to run alongside using its own database schema. Containers isolate deployments of applications in the same SAP HANA platform.

Note:
With XS development, all database objects from all applications were deployed to one giant fixed system schema (called _SYS_BIC). You could not choose your own deployment schema. This meant you could never have multiple versions of the database objects supporting different versions of applications as all updates would overwrite the previous objects as they all lived in the same schema.

Deprecated Interfaces of SAP HANA

SAP HANA Studio

When SAP HANA was launched in 2011, SAP HANA Studio was the main interface that was provided. SAP HANA Studio was aimed at multiple roles and provided tools for database administrators, application developers, and data modelers. SAP HANA Studio has now been replaced with newer web-based interfaces.

However, SAP HANA Studio is still needed by customers who need to maintain SAP HANA applications that were built in the older development and runtime environment (XS Classic).

This is because you cannot maintain applications that were built on XS Classic using the newer interfaces.

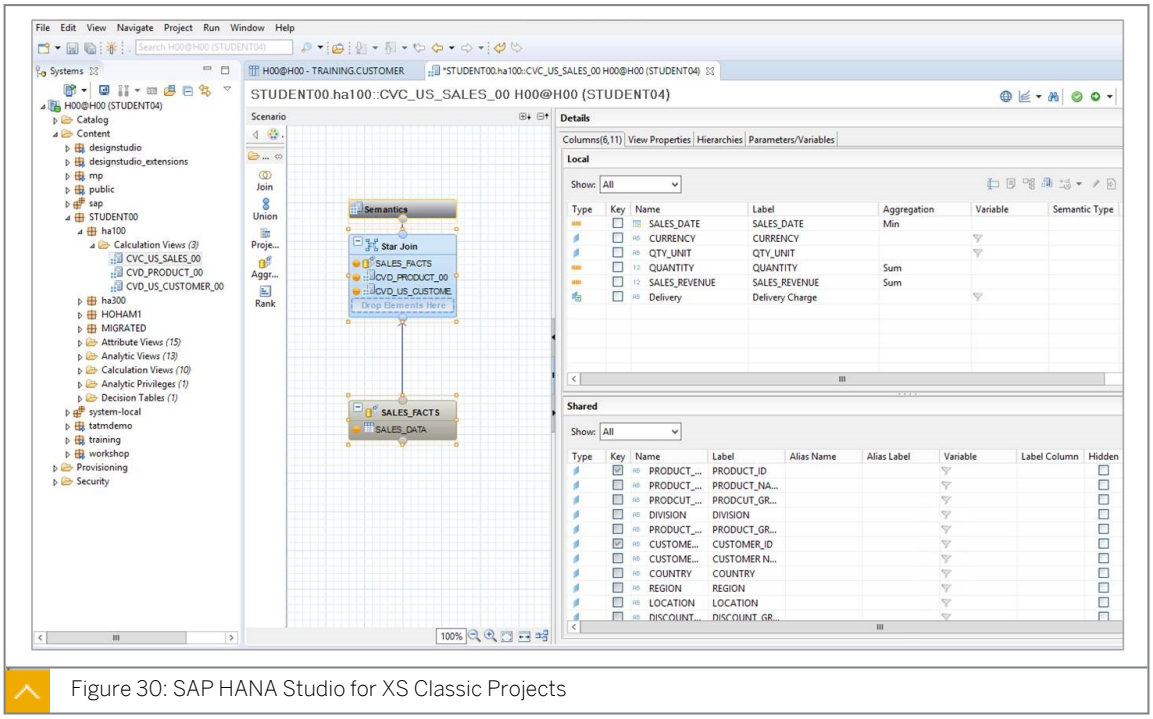


Figure 30: SAP HANA Studio for XS Classic Projects

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 plug-ins provided by SAP to support SAP HANA development and administration.

SAP HANA Studio is no longer developed and its use will reduce over time until it is completely unnecessary. All new innovations are only available in the newer interfaces.

The newer interfaces are *Web IDE for SAP HANA* (for developers) and *SAP HANA Cockpit* (for administration and monitoring) which cover the existing features of the SAP HANA Studio. So there is no reason to use SAP HANA Studio unless you want to maintain objects from XS Classic applications.

SAP HANA Studio cannot be used to develop and maintain XS Advanced applications.

SAP HANA Web-Based Development Workbench

Shortly after the launch of SAP HANA Studio, SAP introduced the *Web-based Development Workbench* to provide application developers with a light-weight tool that could replace SAP HANA Studio.

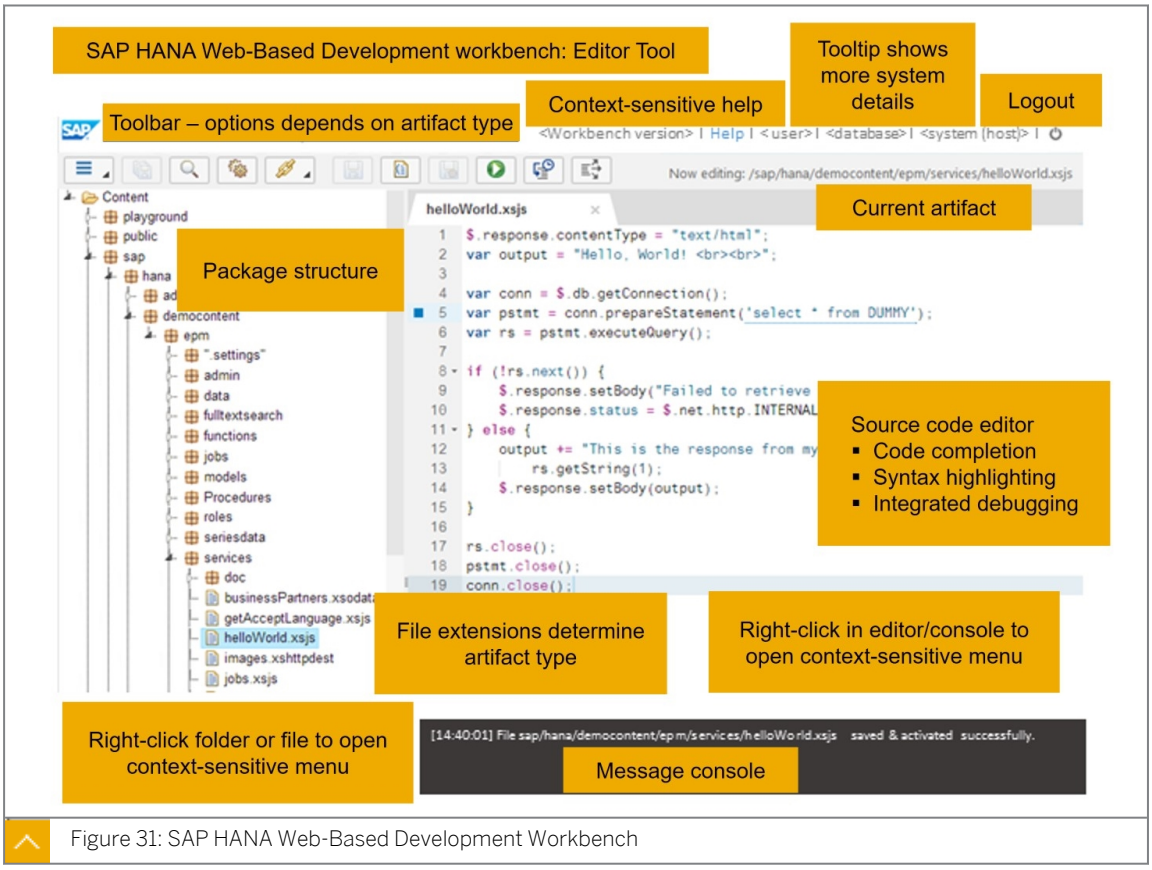


Figure 31: SAP HANA Web-Based Development Workbench

The *Web-based Development Workbench* provides only tools that a developer needs. There are no tools provided for installation, administration, or monitoring. SAP HANA Studio was still needed for those tasks.

Just as with SAP HANA Studio, the Web-based Workbench only supports XS Classic applications and so it is of no use for modern XS Advanced Applications. Unless you are supporting older XS Classic applications, you should not use the Web-based Workbench and instead use Web IDE for SAP HANA.

Caution:
The Web-based Workbench is often mistakenly referred to as the *Web IDE* by many people. This is incorrect as the Web IDE is a completely separate interface.

LESSON SUMMARY
You should now be able to:

- Describe the key technologies of SAP HANA

Deploying SAP HANA

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Understand deployment options of SAP HANA

Deployment Options of SAP HANA

SAP HANA can be deployed on-premise or in the cloud. Let's highlight key features of both options.

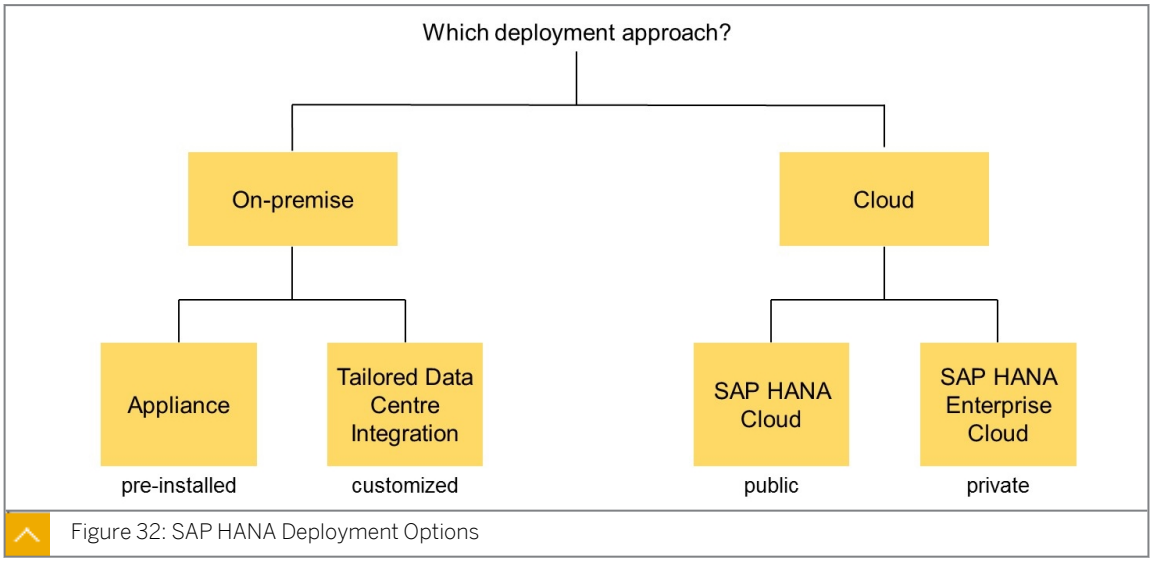


Figure 32: SAP HANA Deployment Options

On-premise Deployment

An on–premise deployment means that SAP HANA is installed in the customer's data center. In this scenario, a customer can choose to install SAP HANA on their own, SAP-approved hardware. This is known as **tailored data center integration**' — or they can choose to have SAP HANA pre-installed and shipped by an approved hardware vendor on the vendor's hardware. This is know as an **appliance**.

The tailored data center integration approach allows customers to mix and match new hardware or re-cycle their own existing hardware to use in their SAP HANA implementation. In this case customers must only use hardware components that are approved by SAP to ensure best performance. This approach requires skilled consultants to make the selections and integrate the components correctly.

The appliance approach provides customers with sizing choices, usually referred to as t-shirt sizes: small, medium, large. Here the memory, disk, CPUs are configured to set sizes. The aim here is to simplify the sizing exercise and provide customers with the assurance that the components work well together.

Benefits of the on-premise approach include having complete control over maintenance and upgrades. Another advantage is that customers can fine-tune the performance of their own deployment as they have complete control of the platform. One of the disadvantages to an on-premise solution is that customers must have their own in-house resources (skilled people, facilities) to run the platform, and they must also purchase or lease their own hardware and licenses.

Cloud Deployment

A cloud deployment means that SAP HANA is installed and runs on hardware that belongs to a dedicated cloud provider. Customers can choose between a **public** cloud provider, such as Amazon Web Services or Google Cloud Platform, or they can deploy to a **private** cloud hosted by SAP. A private, hosted cloud is know as SAP HANA Enterprise Cloud (HEC).

Firstly, there are many benefits to choosing a public cloud provider. These include the speed of deployment — customers can easily deploy a new SAP HANA using a simple, guided tool. Customers can easily increase or decrease the deployed size of their SAP HANA as their requirements change using just a few clicks. Because the customer uses a shared SAP HANA, the costs of deployment are usually lower than with other deployment approaches. Finally, another key benefit of the public cloud is that new features are available to the customer faster than with the other deployment approaches.

The public cloud platform is managed by SAP or other well-known hyper scalers such as Amazon, Google and Microsoft. The advantage of the public 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 teams to install and 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. With the public cloud option, you can change your configuration size anytime, so as you applications grown you can purchase more, or even go smaller if you find you are not using your full capacity. SAP HANA is paid for by subscription or by usage. It is considered to be a service, just like a utility such as gas or water. Another advantage is that SAP provide new features very quickly in an incremental fashion so customers can take advantage of new innovations in their applications.

One of the disadvantage of a public cloud solution is that SAP controls the maintenance and upgrade schedules, so customers have no control over this schedule. Thus, they have to follow SAP's release schedule, which tends to move at a brisk pace to deliver new innovations as soon as they are available. This might not be ideal if the customer prefers to control their own maintenance and upgrade schedule, especially if the customer is constantly developing their own add-on applications and needs a more stable platform on which to build.

Also, the public cloud deployment may not be the best solution if a customer runs a mission critical application and needs to be able to control and fine-tune the resources.

A private cloud approach means that the customer's SAP HANA is hosted by SAP, or an SAP partner but unlike the public cloud, the private cloud platform is not shared with other customers. This means that the customer maintains control of the upgrades to SAP HANA and does not have to follow the public cloud schedule. It also means that the customer is able to control the tuning of the deployment working with SAP or hosting partners to choose the best configuration. However, a private cloud comes at higher cost than with public cloud as the platform is used exclusively by one customer.

Hybrid Approach

Another popular option is a hybrid approach, where a combination of on-premise and cloud is used. For example, a customer may want to have an on-premise deployment of SAP HANA to run their on-premise ERP solution. However, they are also developing new, innovative applications that require more advanced SAP HANA services and infrastructure than they

currently have in-house. Because of this, they use a SAP HANA Cloud to develop and run these new 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 types of deployment in a single application. Since SAP HANA 2.0 was introduced, it is possible to develop applications that can easily be deployed for on-premise and then later moved to the cloud (and the other way around).

SAP HANA On-Premise

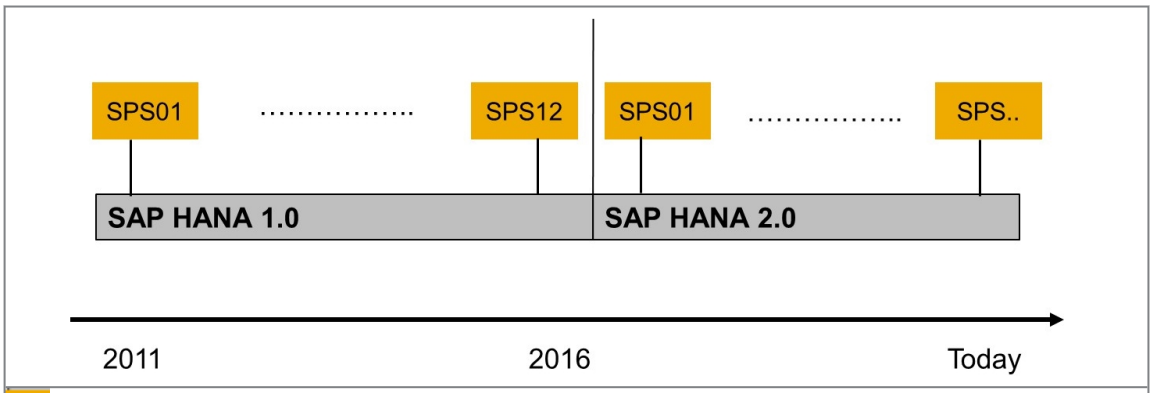


Figure 33: Timeline of SAP HANA Versions

SAP HANA was introduced in 2011 and at that time was available only as an on-premise deployment. There was no cloud deployment option available.

On-premise means that the entire solution is managed by the customer. This includes the installation of the solution on the customer's hardware and the running of the solution that will include such tasks as back-ups, restarts, upgrades, and so on. This is an appealing approach for customers who like to take ownership of their solutions and control the updates and fine-tuning.

The full name of the solution was SAP HANA 1.0. Every six months SAP released a new Support Pack Stack (SPS) that provided new functionality. The support packs started with SPS01 and continued until the delivery of the final support pack stack which was called SPS12. So when you refer to a specific version of SAP HANA you must always add the SPS number to the end, for example, SAP HANA 1.0 SPS12 so that you refer to a precise version of SAP HANA.

In 2016, SAP introduced the next major release of SAP HANA and it was given the full name SAP HANA 2.0. Again, the support pack stacks continue to be delivered but now only every 12 months. This slower pace of enhancement introduction means less disruption caused by system down-time for upgrades, re-testing effort etc. For SAP HANA 2.0 the support pack stack number started again from SPS01.

In between the release of the support pack stacks, SAP provides smaller, incremental updates to the individual components of SAP HANA. You should take note of the exact version of each component. For example the SAP HANA database has a *revision* number. The Web IDE also has a *build* number, and so on. Any urgent fixes are provided to customers quickly through these updates to components.

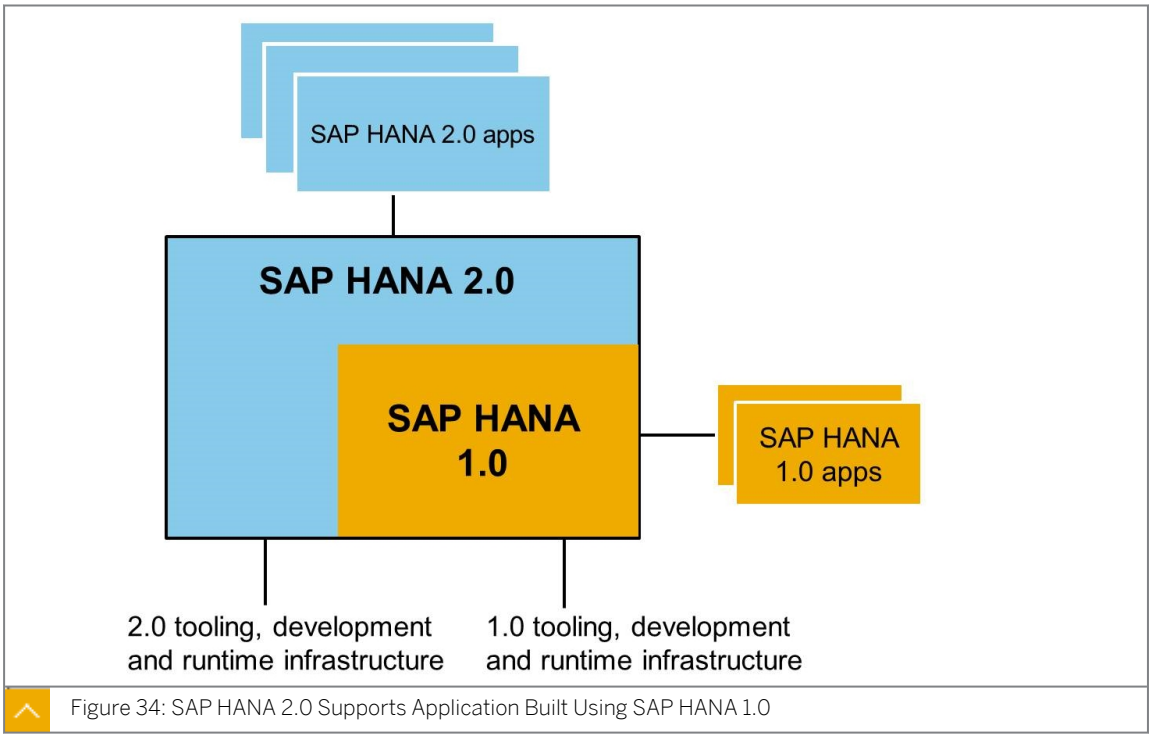


Figure 34: SAP HANA 2.0 Supports Application Built Using SAP HANA 1.0

Applications that were built to run on SAP HANA 1.0 can continue to run unchanged on SAP HANA 2.0. This is possible because SAP HANA 2.0 includes all design-time and runtime capabilities of SAP HANA 1.0. This approach was followed to ensure compatibility with applications that continue run on the technology of SAP HANA 1.0 whilst delivering an improved platform to develop more powerful applications under SAP HANA 2.0 technology. SAP did not want to have customers implement two versions of SAP HANA, side-by-side. Only one implementation of SAP HANA is needed to run application that are developed using versions 1.0 and 2.0.

There are some significant differences between 1.0 and 2.0 versions, and applications that are built using the newer tooling and design and runtime infrastructure of SAP HANA 2.0 will not run under SAP HANA 1.0.

SAP provides migration tooling to support the conversion of applications that were built using SAP HANA 1.0 to native SAP HANA 2.0 applications. Migration of applications is recommended in order to take advantage of the new features and tooling provided with SAP HANA 2.0.

SAP HANA 1.0 technology is deprecated and will no longer developed by SAP. Support continues for now in order to ensure customers, who run SAP applications that are based on SAP HANA 1.0 technology, can continue to run those applications.

Note:
This course is based on SAP HANA 2.0.

SAP HANA Cloud

SAP HANA Cloud is a public cloud service offering of SAP HANA. SAP HANA Cloud is not simply SAP HANA on-premise in the cloud, but a fully-native cloud service that offers nearly all the features of on-premise SAP HANA, plus some additional unique services.

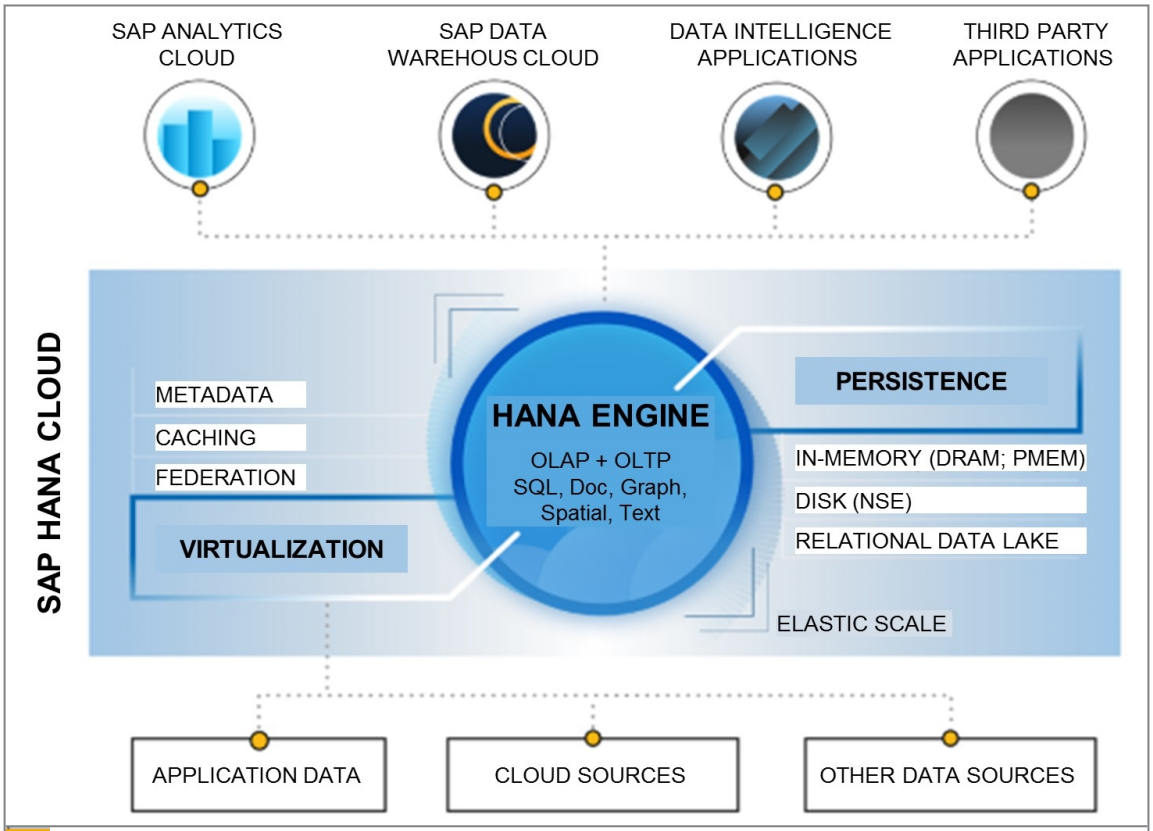


Figure 35: SAP HANA Cloud

SAP HANA Cloud appeals to customers who do not want to commit resources to installing and running SAP HANA. Also, they do not want to own or lease their own infrastructure (hardware, networks, and so on.).

SAP HANA Cloud can be deployed in a few minutes using a web browser. During deployment, choices are made relating to the size of computing powers (CPU) and data storage and any additional cloud services that might be needed. These choices can be changed at a later date.

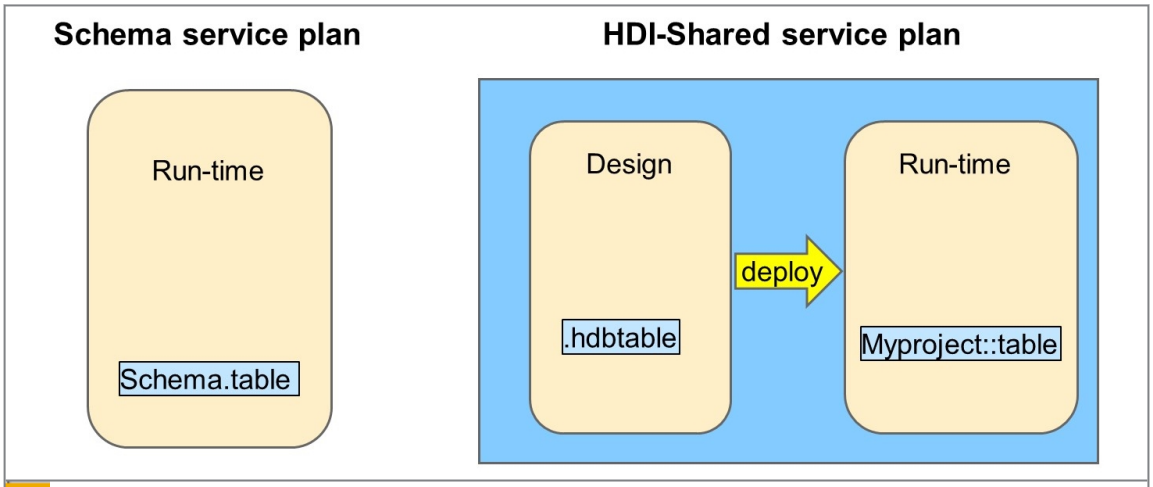


Figure 36: SAP HANA Cloud service plans

SAP HANA Cloud provides two choices of service plans:

Schema service plan — Create all database artifacts directly to schemas using SQL. This run-time only option is the approach that has been used for many years with traditional

databases and appeals to organizations that only need a database to power their applications and do not need design-time tooling.

HDI Shared service plan — Provides a complete design and runtime environment. It uses containers for database isolation. This plan provides tools for the full life-cycle development of database artifacts and deployment tools.

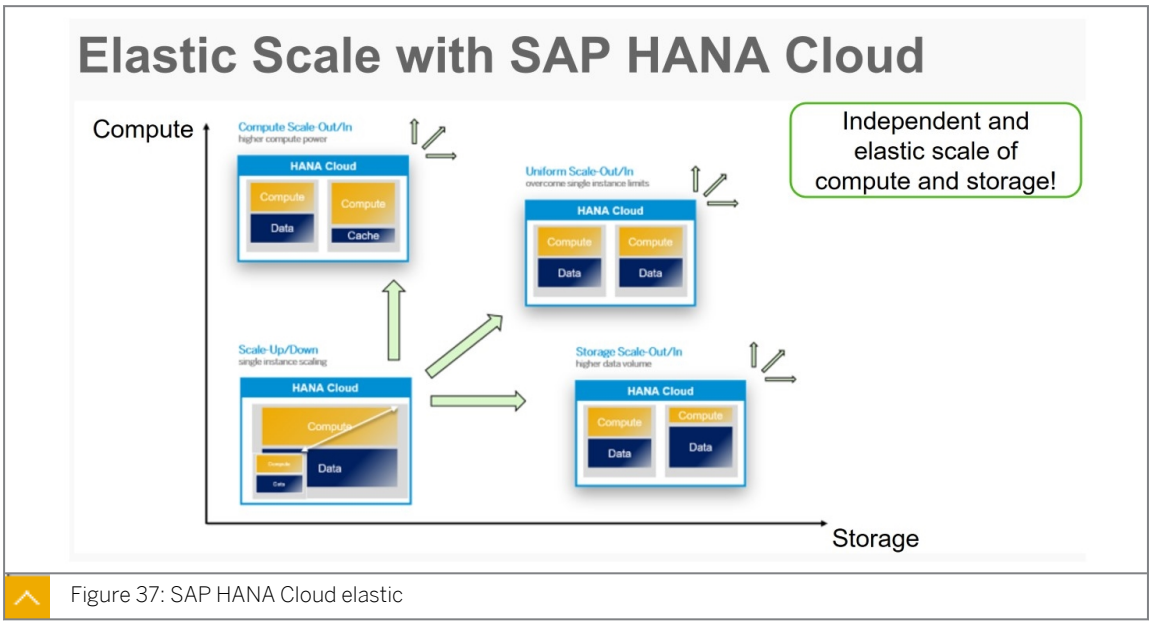


Figure 37: SAP HANA Cloud elastic

One of the most appealing aspects of SAP HANA Cloud is its **elasticity** regarding compute and data storage. Customers can easily increase or decrease their compute and/or data storage using a simple interface.

The elasticity of SAP HANA Cloud is very appealing to SAP HANA on-premise customers who can use SAP HANA Cloud to provide additional, sometimes temporary, compute and storage resources to extend their on-premise landscape.

During SAP HANA Cloud deployment, you can select a data lake service. A data lake service provides data storage for warm data which is stored on a disk based layer. This is ideal for storing large amounts of data where performance does not need to be as optimal as in-memory storage.

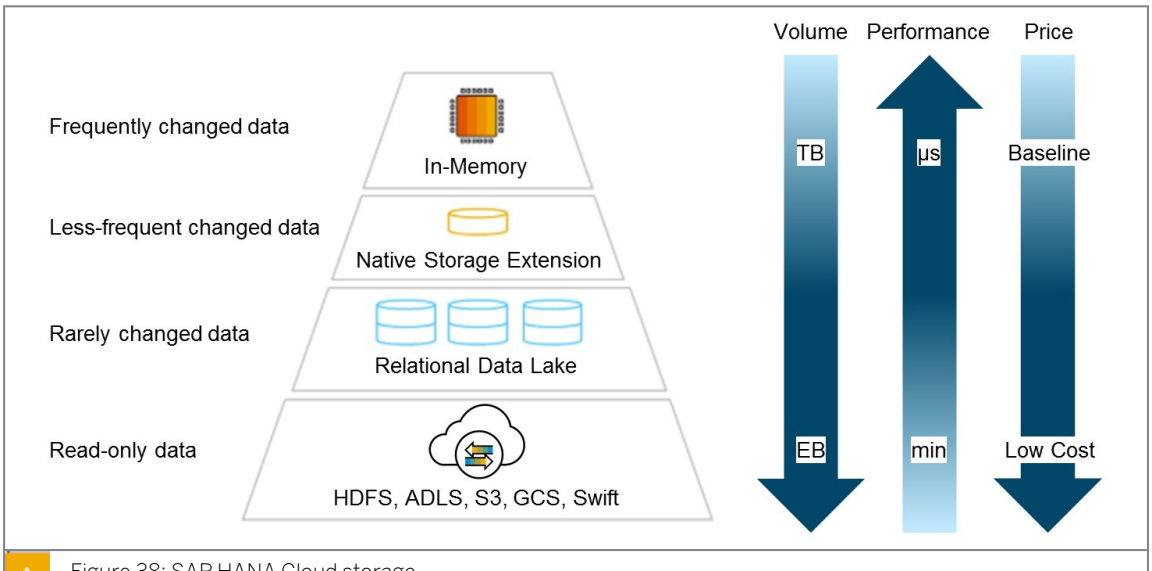


Figure 38: SAP HANA Cloud storage

SAP HANA Cloud receives very frequent updates, so new features are added usually every few weeks. Whilst this might sound appealing to some customers who want to develop applications using the latest technologies, it is also not attractive to those who require a more stable environment to run mission critical applications. This is why SAP provide a hybrid approach.

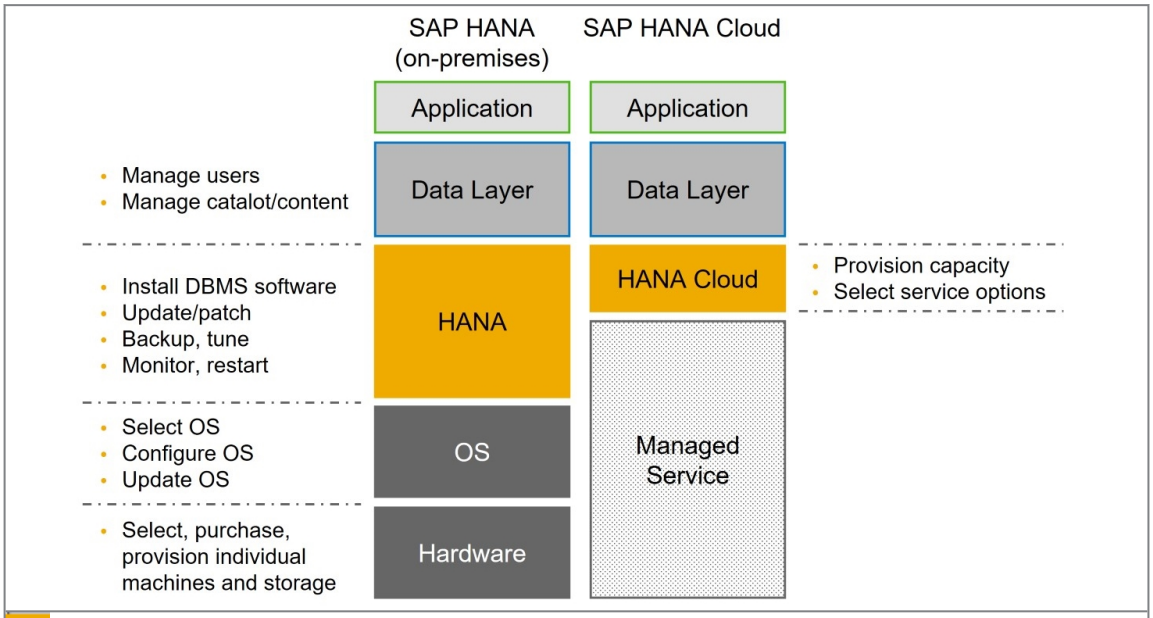


Figure 39: SAP HANA Cloud vs. on-premise

The majority of organizations already have a hybrid cloud strategy which means they see value in having on-premise deployments integrated with cloud deployments of applications.

The appeal of a hybrid strategy is being able to have the security and stability that comes with on-premise deployments for running mission-critical applications. Added to this the flexibility and agility in adjusting resources and using the very latest technology to meet the needs of new business demands of modern, innovative applications with cloud deployments.

HANA Enterprise Cloud

SAP HANA Enterprise Cloud is a private, fully managed cloud designed for mission-critical SAP enterprise solutions, such as SAP S/4HANA, SAP Business Suite and SAP BW/4HANA.

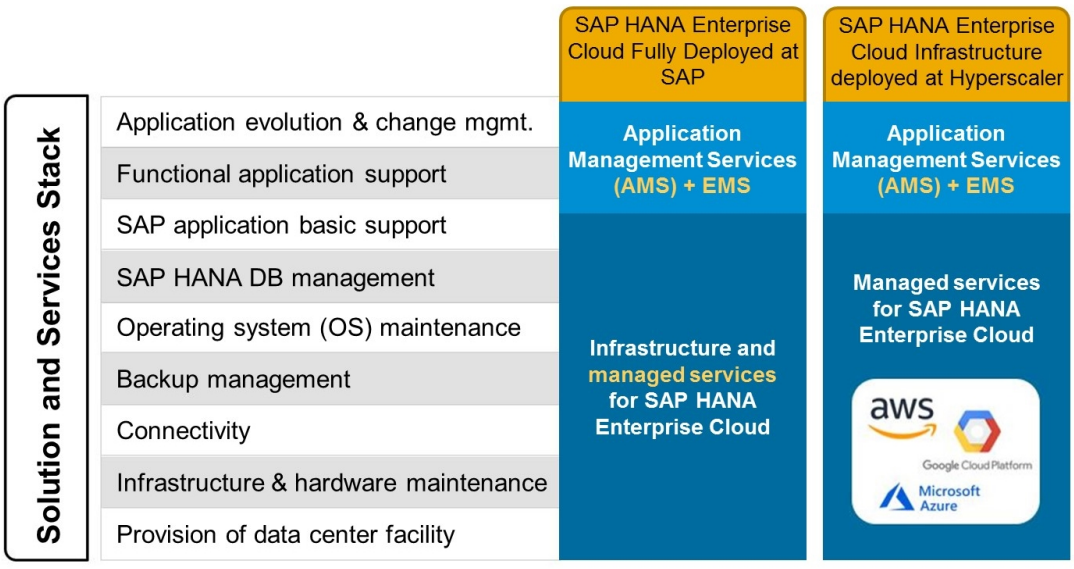


Figure 40: SAP HANA Enterprise Cloud

Customer's choose SAP HANA Enterprise Cloud so that they can focus on their own businesses and not have to manage the IT around their SAP business solutions.

SAP HANA Enterprise Cloud provides services for both the **application layer** and also the **infrastructure layer** . This is often referred to as a full-stack service.

SAP HANA Enterprise Cloud managed services are provided under a Service Level Agreement (SLA) and this encompasses the entire stack so that any system issues that overlap the application and infrastructure layers are dealt with by the same support organization. Often when the application and infrastructure layers are managed by separate organizations, it if easy to get into an unhelpful finger-pointing situation which can cause delays to issue resolution.

SAP HANA Enterprise Cloud is provided via a subscription basis or customers can bring their own licences for SAP applications that are either new or were once deployed on-premise and are moving to the cloud.

HEC managed services include backups, patching, provisioning and upgrades, restore and recovery, infrastructure monitoring, and event detection — all in a private cloud environment

LESSON SUMMARY

You should now be able to:

- Understand deployment options of SAP HANA

Unit 1
Lesson 5

Identifying the Key Roles in an SAP HANA Implementation

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

- Describe the Key Roles of an SAP HANA Implementation

SAP HANA Administrator Role

The SAP HANA Administrator is responsible for the installation and on-going maintenance of the SAP HANA Platform. This is typically the responsibility of an organization's IT team.

The role of the SAP HANA Administrator

- Install and ensure continuity of SAP HANA platform
  - Plan the SAP HANA implementation (sizing, deployment options, failover …)
  - Installation and updates of SAP HANA components
  - Migrate legacy databases to SAP HANA
  - Monitoring and troubleshooting SAP HANA
  - Disaster recovery
- Key skills required:
  - SAP HANA core / advanced administration and technology (HA200 / HA201)
  - SAP HANA performance tuning (HA215)
  - Basic SQL knowledge
  - Might be needed: migration to SAP HANA, tools and processes (HA250)
- Key SAP HANA tools used:
  - SAP HANA Cockpit
- Collaborates with:
  - application developers, regarding application resource requirements
  - security experts, to implement security settings
  - modelers, to understand the resource requirements for data models and help with performance tuning
  - data architects, to understand the data growth, locality and management

Figure 41: Administrator Role

Administrators are the technicians of SAP HANA and they play a key role in the design and planning of the SAP HANA implementation. They are also responsible for the continued updates and patches to the platform.

Note:
The SAP HANA administrator is often referred to as a basis consultant.

If a migration from a legacy database to SAP HANA is required, the administrator should develop skills around the tooling available to automate and ensure a successful migration.

A key responsibility of the administrator is the ensure that SAP HANA is fully available. This means developing and implementing a fail-over approach. This might include the setting up of redundant backup servers (scale-out).

SAP HANA provides many tools for monitoring and tracing. Administrators should learn how to use these tools and also to setup alerts so that the system can be largely self-monitored.

SAP HANA runs on the Linux operating system and so Linux knowledge is useful.

If the deployment option chosen is SAP HANA Cloud then the role of the administrator is significantly reduced compared to the on-premise deployment. This is because the installation of SAP HANA is not required. SAP HANA Cloud is deployed using a simple interface and requires no IT skills. But the IT team will certainly be involved with sizing the required cloud tenant, and if required, the connection of on-premise and other cloud applications to SAP HANA Cloud. But the day to day administration tasks of keeping SAP HANA running, plus backups are the responsibility of SAP or hosting partners.

SAP HANA Modeler Role

The SAP HANA data modeler is responsible for building the virtual data layer in SAP HANA database. This is built mainly from calculation views but can also include other database artifacts such as functions and procedures. The most important skill of a data modeler is calculation view development. A simple calculation view is easily created but it takes a lot of experience and training to understand how to develop high-performance yet easy to maintain models that exploit the full potential of the SAP HANA in-memory database.

The role of the SAP HANA Modeler

- Develop and maintain high performance data models
  - Calculation views
  - Text
  - Graph
  - Spatial
  - …
- Key skills required:
  - SAP HANA modelling (HA300 / HA301)
  - SQL development (HA150)
  - …
- Key SAP HANA tools used:
  - Web IDE for SAP HANA
- Collaborates with:
  - application developers regarding analytics requirements of applications
  - security experts to design business user access
  - administrators to understand the technical infrastructure options that support anaytics
  - …

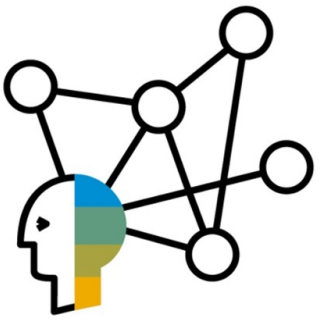


Figure 42: Modeler Role

SAP HANA data modelers should also be familiar with SQL as they will be using this language to develop custom logic to extend the standard capabilities of the calculation views. SQL is also used to test and trace the execution of the calculation views, so again, this role requires a solid understanding of SQL.

Beyond core modeling using calculation views, data modelers should also develop skills in the advanced SAP HANA data modeling topics such as text, spatial, predictive and graph data processing. SAP HANA is a powerful data processing platform and a data modeler who is able to develop sophisticated models that exploit all types of data is a valuable member of an SAP HANA implementation team.

The data modeling role often overlaps with the role of an application developer because most applications require some sort analytical processing. The analytic processing part of an application is best handled by developing and consuming data models using the provided SAP HANA database artifacts such as calculation views and SQL based models, instead of handling the data processing logic within the application code. If the developer is not able to develop such data models they will need to enlist the support of the dedicated data modeler.

Security might also be an area of overlaps with the team members who take care of this part of the implementation. Protecting the data from unauthorized access can be seen as a job for the data modeler who builds database artifacts called Analytic Privileges to provide the rules for data access for each user/role. However, this might also be the responsibility of the security experts in the team. But whoever takes ownership of this task, collaboration is essential between the security experts and data modelers.

Members of an SAP S/4HANA or SAP BW/4HANA implementation team should also develop skills in SAP HANA data modeling so that they can fully exploit the analytical capabilities of these enterprise applications.

SAP HANA Application Developer Role

The main reason most customers choose to implement SAP HANA is because it is offered as an alternative database for their SAP application, for example, SAP Business Suite. Or, it is because the SAP application only runs on SAP HANA.

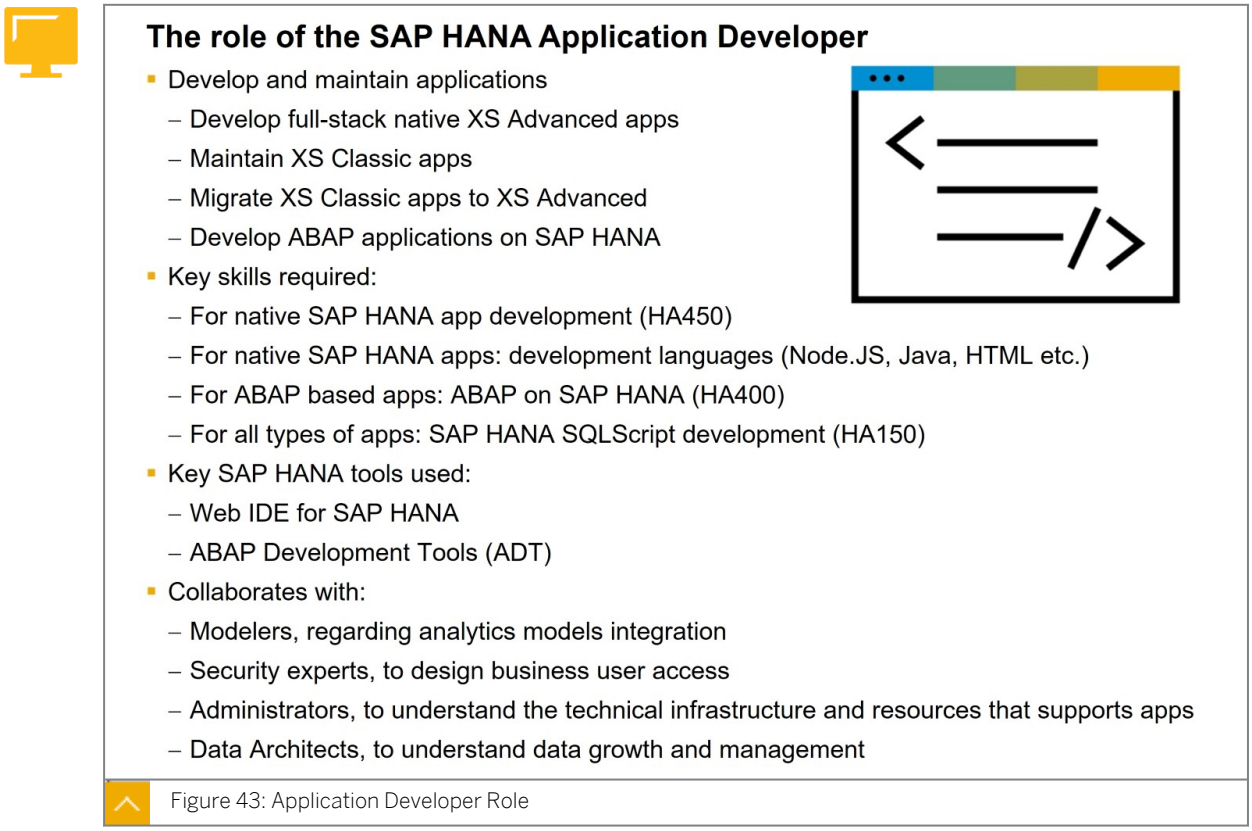


Figure 43: Application Developer Role

SAP applications usually allow for customer extensions and enhancements. For ABAP applications such as SAP Business Suite and SAP S/4HANA, the skills of an ABAP develop are needed. But if the ABAP application is running on SAP HANA, then the developer will need to learn how to write code that is optimized for SAP HANA. An ABAP developer might be surprised to know that they need to re-learn some of the coding techniques when their extensions and other code based artifacts are running on SAP HANA.

Apart from ABAP applications, SAP HANA can host applications that are written completely on the platform. These are referred to a full-stack applications because they require development of objects at each level of the stack: interface, application and database. Many customers are keen to develop or trial new, innovative applications to support their enterprise applications (such as ERP). SAP HANA provides all the tools for the developer to achieve this. but of course, development skills are needed.

The skills needed depend on the languages chosen for the application development. But the base language of SAP HANA database is always SAP HANA SQLScript. Java or Node.JS might also be needed as well as HTML and Core Data Services (CDS)

SAP HANA Security Role

The security role is often taken over by the administrator. But as rules tighten around data privacy and data access, organizations employ dedicated experts to take on this role.

The role of the SAP HANA Security Expert

- Plan and implement security across SAP HANA
  - Authentication (logging on and validation of users)
  - User authorization (data and application access)
  - Compliance and governance (data privacy, data retention, usage)
  - Security tracing, monitoring and auditing
- Key skills required:
  - SAP HANA basic technology and infrastructure knowledge (HA200)
  - SAP HANA detailed technical security skills (HA240)
  - SAP HANA detailed user security skills (HA940)
- Key SAP HANA tools used:
  - SAP HANA Cockpit
  - Web IDE for SAP HANA
  - XS Advanced Cockpit
- Collaborates with:
  - application developers, to understand application and data access rules
  - data architect, to implement data access rules
  - administrators, to jointly agree and implement security settings and policies
  - modelers, to establish data access to models including data privacy rules

Figure 44: Security Role

Security in SAP HANA can be broken up into two parts:

- Technical (platform) security: encryption, firewalls and so on.
- User security: user and role management, privileges assignments.

All SAP HANA implementations will require some skills around security. Often security is mostly regarded as topic for the application, for example, ABAP authorizations, but remember the database also need protection from unauthorized access as it guards sensitive and valuable data.

SAP HANA Data Architect Role

As organizations acquire more data than even before, and data driven applications become the norm, data management becomes an major topic. There are many tools provided by SAP HANA to manage data and this has now become a recognized role in an SAP HANA implementation.

The role of the SAP HANA Data Architect

- Plan, implement and manage a sustainable data architecture
  - Implement data tiering (hot, warm, cold)
  - Data integration
  - Data quality
- Key skills required:
  - SAP HANA technology and infrastructure
  - Data Provisioning tools (HA550)
  - Basic SQL
- Key SAP HANA tools used:
  - Smart Data Access (SDA)
  - Smart Data Integration (SDI)
  - Web IDE for SAP HANA
  - SAP HANA Cockpit
- Collaborates with:
  - application developers, regarding data access requirements and data growth and archiving
  - security experts to agree data access, privacy and security
  - Administrators, to understand and develop the technical infrastructure that supports data management
  - Modelers, to understand the data access requirements of models

Figure 45: Data Architect

In the past, IT would take the role of data architects, providing storage capacity as needed and implementing archiving solutions. But the data requirements of applications have become more sophisticated and we expect data from all sources in real time in addition to access to Big Data.

SAP HANA provides tools for data integration and cleansing. These requires specialist skills that other members of the SAP HANA project may not have.

The data architect learn to work with data management tools that are native to SAP HANA (SDI, SDA, NSE and so on) and also tools that can integrate with SAP HANA (SAP PI, SAP Data Services, SAP LT and so on.)

LESSON SUMMARY

You should now be able to:

- Describe the Key Roles of an SAP HANA Implementation

Learning Assessment

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

   Choose the correct answers.

   - [ ] A Increase in connectivity of people and devices
   - [ ] B Separation of transactional processing and analysis processing to dedicated servers
   - [ ] 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. SAP HANA provides excellent performance for data intense applications because it stores pre-aggregated data for fast access to results.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

4. What are advantages of column store tables versus row store tables?

   Choose the correct answers.

   - ☐ A Data footprint is automatically reduced through compression.
   - ☐ B Only the columns required for processing are actually loaded to memory.
   - ☐ C Columns can be partitioned to improve performance.
   - ☐ D They are optimized for fast writes and updates to records.

5. Why do we still need a disk layer when SAP HANA is referred to as an in-memory database?

   Choose the correct answers.

   - ☐ A To store data that has been unloaded from memory when memory becomes full.
   - ☐ B To hold the delta store for newly-arrived records.
   - ☐ C To enable full database recovery in the event of a power failure.
   - ☐ D To store data that is frequently used.

6. 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.

7. Active/active-read enabled mode allows applications to redirect read-intense workloads to a secondary redundant read-only HANA node?

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

8. Which tables are automatically compressed?

   Choose the correct answer.

   - ☐ A Row
   - ☐ B Virtual
   - ☐ C Column

9. Why might customers choose to implement SAP HANA Cloud?

   Choose the correct answers.

   - ☐ **A** Reduced administration effort on customer side
   - ☐ **B** Easy to scale
   - ☐ **C** Fast deployment
   - ☐ **D** Manage updates to the platform

10. What is meant by SAP HANA Tailored Data Center Integration?

    Choose the correct answer.

    - ☐ **A** The customer creates a customized hardware platform on which to install SAP HANA, by reusing existing and SAP certified hardware components.
    - ☐ **B** Partners provide cloud infrastructure on which SAP HANA can be installed.
    - ☐ **C** Partners provide a fully installed SAP HANA on certified hardware for use on-premise.

11. Which are key roles of an SAP HANA project?

    Choose the correct answers.

    - ☐ **A** Change management consultant
    - ☐ **B** Application developer
    - ☐ **C** Data architect
    - ☐ **D** Security expert
    - ☐ **E** Modeler

Learning Assessment - Answers

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

   Choose the correct answers.

   - [X] A Increase in connectivity of people and devices
   - [ ] B Separation of transactional processing and analysis processing to dedicated servers
   - [X] C Increase in use of mobile devices
   - [X] D Massive growth in data volume

   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 copies of the data from one system to another: More and more people and devices are becoming connected (think about IoT). Mobile device usage is on the increase and also the volume of data continues to rise at a staggering rate.

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

   Correct — We are moving away from disk access and towards memory based data storage and processing using multi-core processors. Memory is now larger and also cheaper that even before and this is a key enabler of the next generation data processing platform.

3. SAP HANA provides excellent performance for data intense applications because it stores pre-aggregated data for fast access to results.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct — SAP HANA does not store pre-aggregated data. The in-memory database is able to aggregated data on the fly from the most granular data and still achieve great performance.

4. What are advantages of column store tables versus row store tables?

   Choose the correct answers.

   ☒ A Data footprint is automatically reduced through compression.

   ☒ B Only the columns required for processing are actually loaded to memory.

   ☒ C Columns can be partitioned to improve performance.

   ☐ D They are optimized for fast writes and updates to records.

   Correct — Column store tables are optimized only for read-intense analytical processing, not transactional processing where not just read but also write to tables is needed. Column store tables also use automatic compression to reduce the footprint and we only need to load columns that are required into memory for processing. Also, you can partition column tables to get better performance when the tables are large.

5. Why do we still need a disk layer when SAP HANA is referred to as an in-memory database?

   Choose the correct answers.

   ☒ A To store data that has been unloaded from memory when memory becomes full.

   ☐ B To hold the delta store for newly-arrived records.

   ☒ C To enable full database recovery in the event of a power failure.

   ☐ D To store data that is frequently used.

   Correct — The disk layer is still 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. The delta store is an area of memory not disk. Memory is always used to store frequently used data to speed up loading to the CPU.

6. 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.

   Correct — A multistore table is a new feature of SAP HANA 2.0, where partitions can be spread across memory and extended storage to support better performance on aged data. It is not some kind of hybrid table that can hold data in row and column orientation. It is not a table that can be shared across multiple instances of HANA.

7. Active/active-read enabled mode allows applications to redirect read-intense workloads to a secondary redundant read-only HANA node?

   Determine whether this statement is true or false.

   - [x] True
   - [ ] False

   Correct — A secondary redundant read-only HANA node is used to continuously capture and replay the database log from a primary HANA node mainly for data recovery scenarios. But the secondary redundant node can now also be used for any read-intense tasks as it is always a direct copy of the primary node. So, off-loading this work from the primary HANA balances the workloads and increase overall performance.

8. Which tables are automatically compressed?

   Choose the correct answer.

   - [ ] A Row
   - [ ] B Virtual
   - [x] C Column

   Correct! Column tables are compressed. Row tables are not compressed and virtual tables never store data.

9. Why might customers choose to implement SAP HANA Cloud?

   Choose the correct answers.

   [X] A Reduced administration effort on customer side

   [X] B Easy to scale

   [X] C Fast deployment

   [ ] D Manage updates to the platform

   Correct — SAP HANA Cloud is managed by SAP and partners and so the administration effort on the customers side is minimal. SAP HANA Cloud is easy to scale because you can choose your own CPU and memory and can change this at anytime very easily. SAP HANA Cloud can be deployed quickly using a simple interface. Customers do not manage the updates, this is done only by SAP.

10. What is meant by SAP HANA Tailored Data Center Integration?

    Choose the correct answer.

    [X] A The customer creates a customized hardware platform on which to install SAP HANA, by reusing existing and SAP certified hardware components.

    [ ] B Partners provide cloud infrastructure on which SAP HANA can be installed.

    [ ] C Partners provide a fully installed SAP HANA on certified hardware for use on-premise.

    Correct — Tailored Data Centre Integration enables customers to reuse their existing hardware. Tailored Data Centre Integration is not a partner provided cloud infrastructure or even a fully installed SAP HANA on certified hardware to use on-premise.

11. Which are key roles of an SAP HANA project?

    Choose the correct answers.

    [ ] A Change management consultant

    [X] B Application developer

    [X] C Data architect

    [X] D Security expert

    [X] E Modeler

    Correct! Key SAP HANA roles include application developer, data architect ,security expert, and modeler. A change management consultant is not a key SAP HANA role.

UNIT 2 Technical Requirements of SAP HANA

UNIT OBJECTIVES

- Understanding the SAP HANA technical deployment options
- Describe high availability
- Describe SAP HANA Lifecycle Management Tools
- Describe the SAP HANA Revision Strategy

Unit 2
Lesson 1

Technical Deployment Options

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

- Understanding the SAP HANA technical deployment options

Technical Deployment Options

Scaling SAP HANA

Scale-up and scale-out are the two general approaches you can take to increase the processing power and storage capacity of your SAP HANA system.

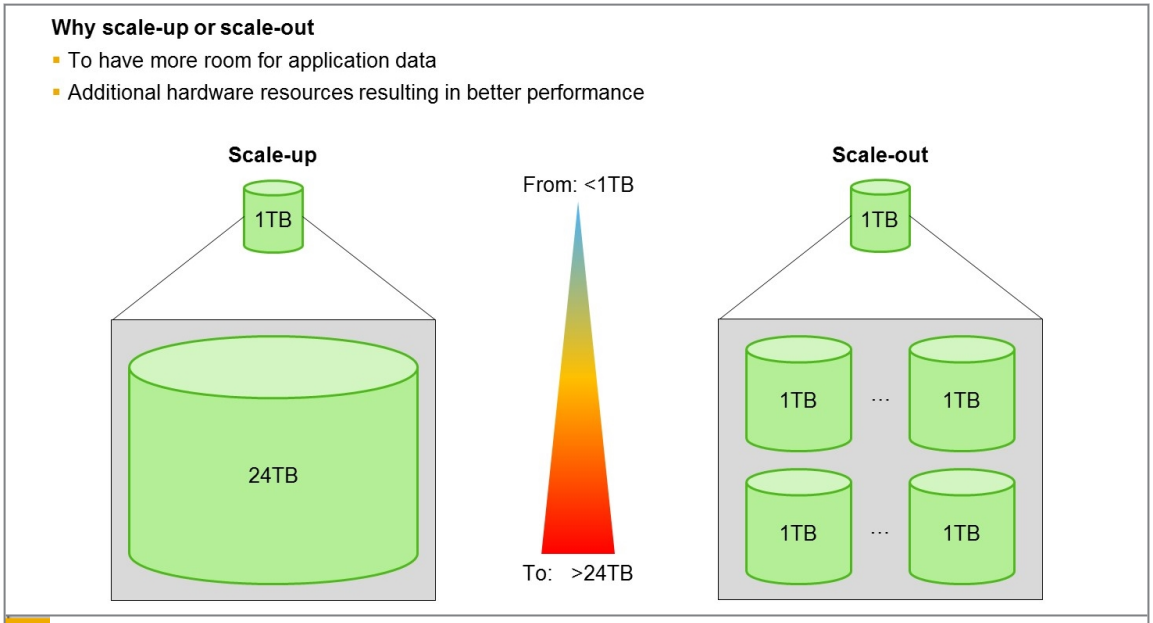


Figure 46: Scale-up or Scale-out

Scale-up means increasing the size of one physical machine by increasing the amount of RAM available for storing and processing data. Increasing the RAM also means increasing the number of CPU's, which means increasing the processing power of SAP HANA.

Note:
The largest supported SAP HANA appliances have up to 24TB of RAM and up to 448 cores divided over 16 sockets.

Scale-out means combining multiple independent servers into one large SAP HANA database system. The main reason for distributing an SAP HANA database system across multiple

hosts (that is, scaling out) is to overcome the hardware limitations of a single physical server. This allows an SAP HANA database system to distribute the load between multiple servers.

In such a distributed system, each index server is usually assigned to its own host to achieve maximum performance. It is possible to assign different tables to different hosts by partitioning the database, as well as to split a single table between hosts by partitioning of a table.

A typical scale-out cluster consists of 2 to n servers per cluster, where in theory there is no limit for n. The largest certified configuration is 112 servers, while the largest tested configuration has more than 250 servers. Each server configuration is either 4 CPU/2 TB or 8 CPU/4 TB.

Both architectures, scale-up and scale-out, provide full support for high availability and disaster recovery.

Single-host or Multiple-host SAP HANA systems

An SAP HANA system can be configured as a single-host or multiple-host system. This configuration is determent during the installation but can easily be changed afterwards.

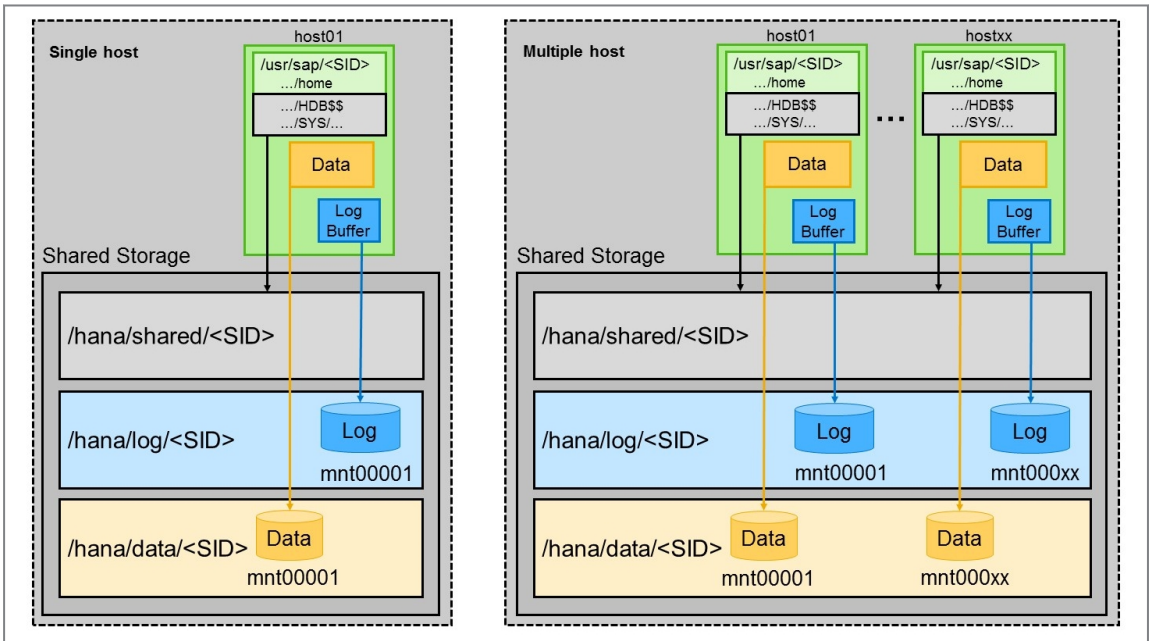


Figure 47: Single-host or Multi-host

Single-host system

A single-host system is the simplest system installation type. It is possible to run an SAP HANA system entirely on one host and then scale the system up as needed. The installation tool called SAP HANA database lifecycle manager can be used to install an SAP HANA single-host system in one of the program interfaces, and with a combination of parameter specification methods.

Multi-host system

A multiple-host system is a system with more than one host, which can be configured as active worker hosts or idle standby hosts. The SAP HANA Database Lifecycle Manager (HDBLCM) can be used to install an SAP HANA multiple-host system. To add hosts to an existing system, use the SAP HANA resident HDBLCM.

SAP HANA Certified Hardware

A typical single-server deployment of SAP HANA can vary from a 2 CPU configuration with 128 GB of RAM as an entry-level system to a high-end 8 CPU socket with244 cores and up to 24 TB of RAM.

The SAP HANA hardware needs to be certified. A list of all the certified SAP HANA hardware can be found in the SAP HANA Hardware Directory.

> Note:
> The SAP HANA Hardware Directory is available here:
>
> `https://www.sap.com/dmc/exp/2014-09-02-hana-hardware/enEN/index.html`

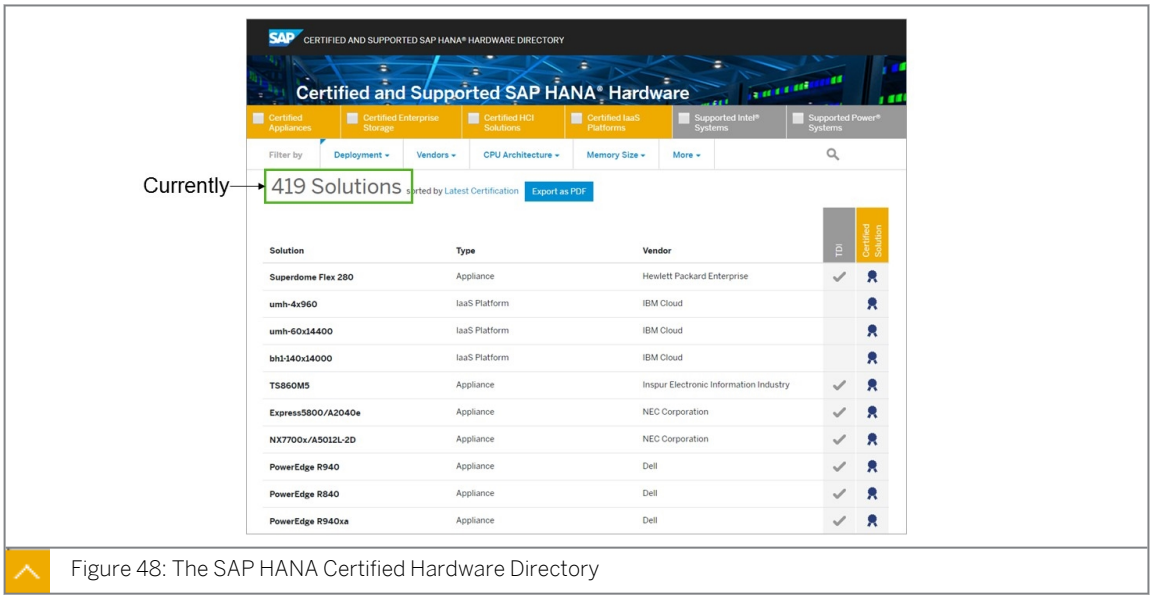


Figure 48: The SAP HANA Certified Hardware Directory

The SAP HANA Hardware Directory lists all hardware that has been certified within the SAP HANA hardware certification program or is supported for a specific SAP HANA SPS releases and Intel® / Power® architecture. The Supported Intel® Systems are valid for specific service packs. The hardware was tested by the hardware partner with SAP LinuxLab. The systems are supported for SAP HANA.

The hardware is required to have a valid SAP HANA Hardware certification at the point of purchase by the customer. Once the validity date of the certification has passed, the hardware will continue to be supported by the Partner until the end of maintenance as indicated by the Partner.

SAP HANA Tailored Data Center Integration

Customers are not restricted in their hardware choices by the list of available SAP HANA appliances sold by our hardware partners. The customer can also decide to build an SAP HANA system by using already available hardware. This approach is called SAP HANA Tailored Data Center Integration (TDI).

Compared to the appliance delivery,in order to optimally integrate SAP HANA in your data center, SAP HANA tailored data center integration provides you with more flexibility regarding the hardware components required to run SAP HANA.

SAP does not offer any formal certification of a given SAP HANA TDI infrastructure. Instead, SAP certifies the server hardware for SAP HANA computing nodes and Storage solutions with SAP HANA hardware partners and requires that customers only use these certified configurations for SAP HANA TDI deployments.

Caution:
SAP requires that the installation of the SAP HANA software is done by persons who passed SAP's certification for SAP HANA installation administrators.

Every SAP HANA TDI infrastructure that fulfills these prerequisites is supported by SAP and does not need any additional approval. Before going productive with an SAP HANA system deployed on a SAP HANA TDI infrastructure.

SAP recommends conducting a HANA Go-Live Check as offered by SAP Digital Business Services.

Sizing SAP HANA Hardware

When sizing an SAP HANA database, the sizing approach depends on the implementation scenario used by the customer. The possible sizing approaches are as follows:

- Customers new to SAP applications and SAP HANA can use the Quick Sizer to size the required memory for an SAP HANA system. This method can also be used for existing SAP customers who want to perform a greenfield implementation.
- Customers who want to migrate their existing SAP NetWeaver based systems to run on SAP HANA can execute a sizing report (/SDF/HDB_SIZING) which calculates the required memory. For the non-SAP NetWeaver based systems you use the sizing method described in SAP Note 1514966.

Note:
The Quick Sizer tool is available here:

`https://www.sap.com/about/benchmark/sizing.quick-sizer.html`

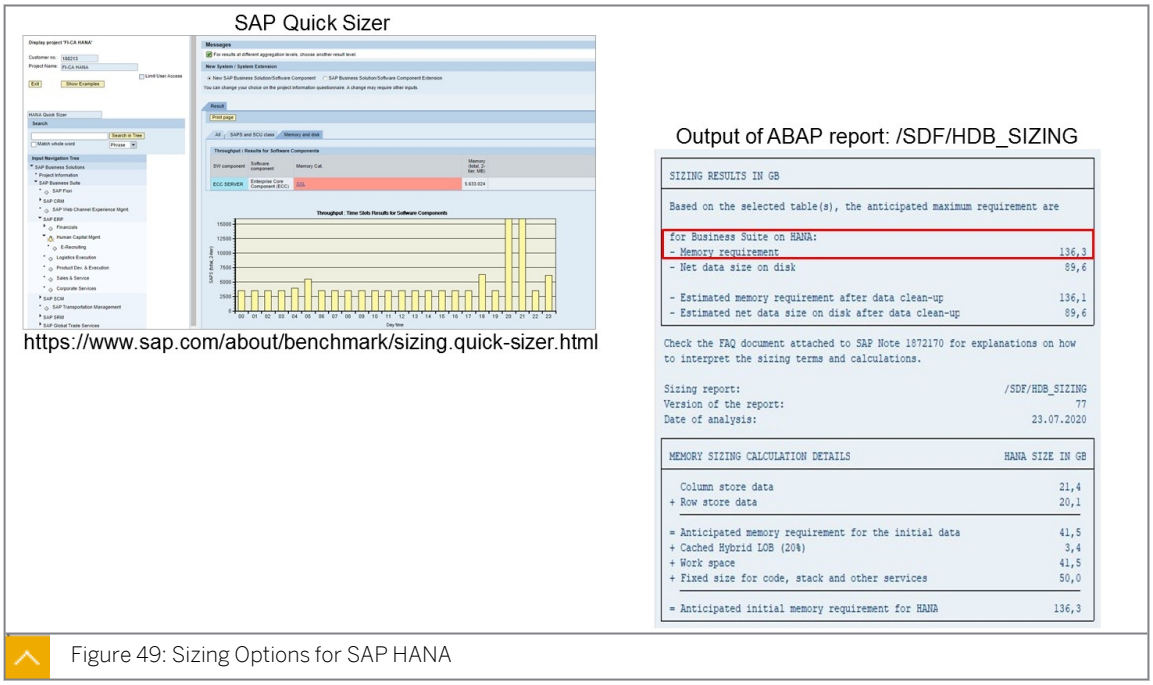


Figure 49: Sizing Options for SAP HANA

When the customer needs to run the SAP HANA database in the on-premise data center, then the SAP HANA can be installed as an appliance solution or a Tailored Data center Integration (TDI) solution. In both cases the customer is responsible for the hardware, the Linux operating system and the SAP HANA database system.

SAP HANA Operating System Requirements

SAP HANA database system is available on the Linux operating system for the distributions from SUSE and Red Hat. Both companies ship optimized for SAP applications. These version also support SAP HANA.

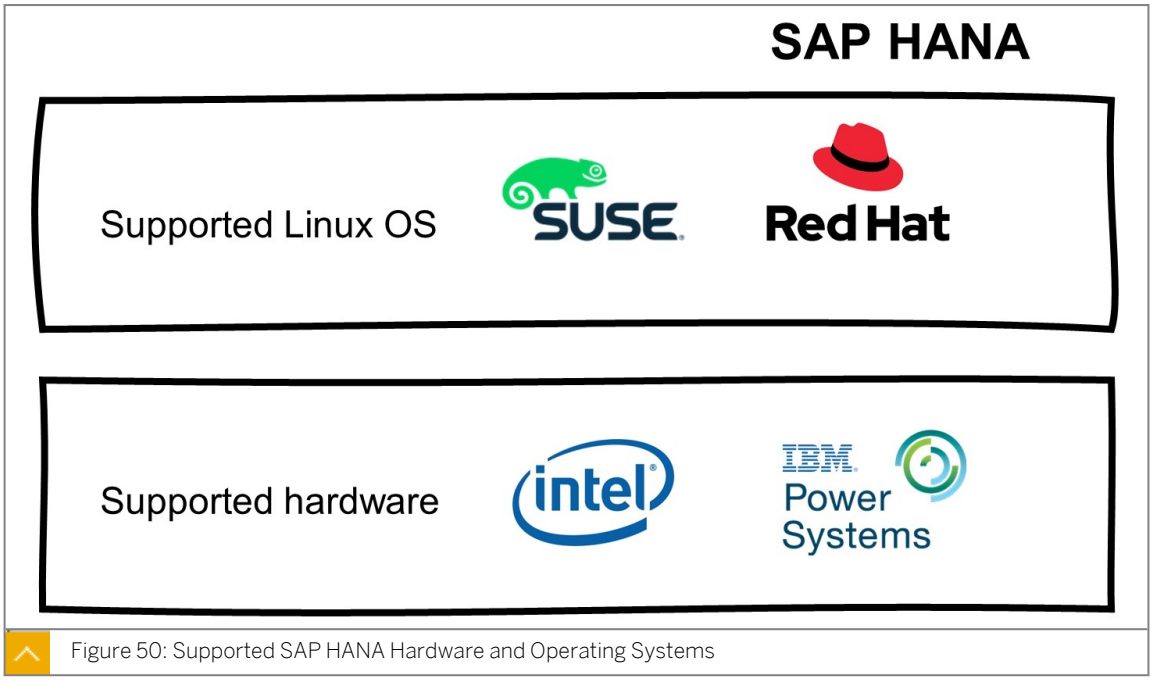


Figure 50: Supported SAP HANA Hardware and Operating Systems

Supported Hardware Platforms

SAP HANA is available for Intel-based hardware platforms and IBM Power Systems. Which hardware configuration are certified for IBM Power Systems and Intel-based hardware platforms can be found on the SAP HANA Hardware Directory.

Supported Operating Systems

SAP HANA is supported on SUSE Linux Enterprise Server (SLES) and Red Hat Enterprise Server (RHEL). Which version on the Linux operating system is required depends on the SAP HANA version that is used. Currently SAP HANA 1.0 and SAP HANA 2.0 are in support.

Supported Operating Systems for SAP HANA 1.0 on Intel-Based Hardware Platforms:

- SUSE Linux Enterprise Server 15 SPx, 12 SPx and 11 SPx
- Red Hat Enterprise Server 7.x and 6x

Supported Operating Systems for SAP HANA 1.0 on IBM Power Servers:

- SUSE Linux Enterprise Server 11 SPx

Supported Operating Systems for SAP HANA 2.0 on Intel-Based Hardware Platforms:

- SUSE Linux Enterprise Server 15 SPx and 12 SPx
- Red Hat Enterprise Server 8.x and 7x

Supported Operating Systems for SAP HANA 1.0 on IBM Power Servers:

- SUSE Linux Enterprise Server 15 SPx and 12 SPx
- Red Hat Enterprise Server 8.x and 7x

Note:
For all the specific operating system version details, see SAP Note 2235581 - SAP HANA: Supported Operating Systems

Real Hardware or Virtualized Hardware

SAP HANA can be deployed on physical hardware or on virtualized hardware. Which option to choose depends on the customers infrastructure situation, but in both situations the hardware needs to be certified.

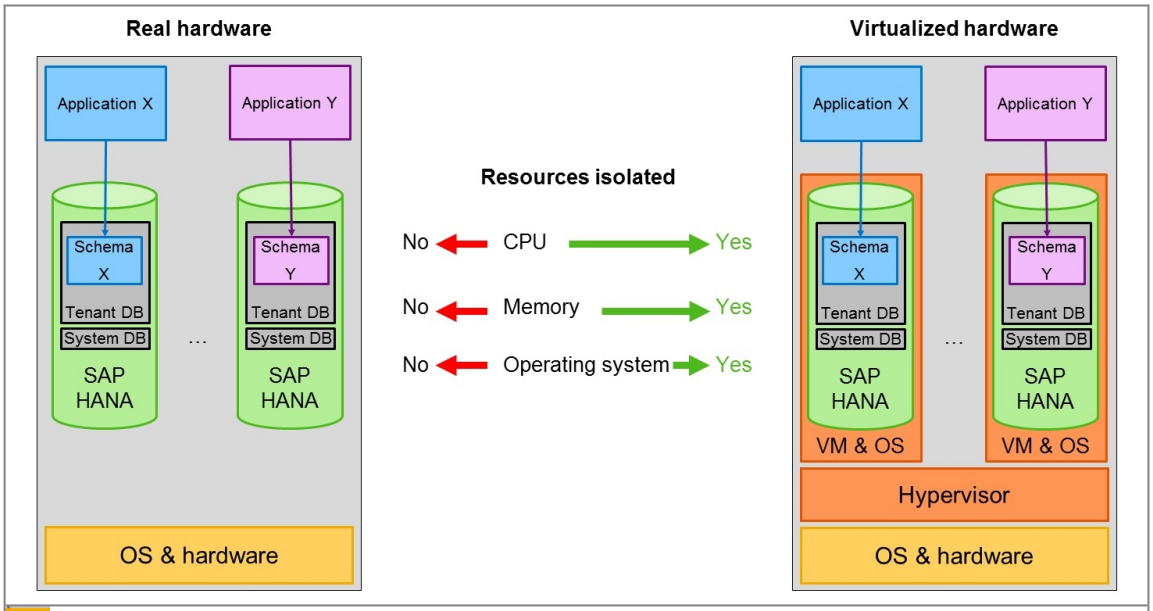


Figure 51: Real Hardware or Virtualized Hardware

SAP HANA and Virtualization

SAP HANA supports many popular, such as VMware vSphere, Hitachi LPAR, Huawei FusionSphere, Red Hat Enterprise Linux KVM Hypervisor, SUSE Linux Enterprise Hypervisor, XEN, Nutanix Acropolis Hypervisor, IBM PowerVM.

Note:
Some of these virtualization technologies are only supported for non-productive environments. For more details on the use cases of the supported virtualization technologies see SAP Note 1788665 - SAP HANA Support for virtualized/ partitioned (multi-tenant) environments.

You can choose virtualization for a wide variety of configurations, for single and multiple virtual machines, in single- or multinode configurations, for appliances of SAP HANA and SAP HANA tailored data center integration delivery methods.

The technical deployment type, SAP HANA with virtualization, refers to the scenario where one or more SAP HANA database systems are deployed on one or more virtual machines running on certified SAP HANA server hardware.

The first benefit of virtualization is that you can assign dedicated CPU and memory resources to specific SAP HANA database systems and by doing this you increase the flexibility and efficiency of hardware usage.

A second benefit of virtualization is that the hypervisor is virtualizing the complete Linux operating system. In this way you can run multiple SAP HANA database systems completely isolated from each other on the same hardware.

In a virtual environment, you can still run multiple SAP HANA database systems in one virtual machine, having multiple tenant databases and schemas. A virtualized SAP HANA offers the same possibilities as an SAP HANA on real hardware, except that by virtualizing the SAP HANA database the total cost of ownership is reduced.

LESSON SUMMARY

You should now be able to:

- Understanding the SAP HANA technical deployment options

Unit 2
Lesson 2

High Availability and Disaster tolerance

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

- Describe high availability

SAP HANA and Managing Server Failure

SAP HANA is fully designed for high availability. It supports recovery measures ranging from faults and software errors, to disasters that decommission an entire data center. High availability is achieved by eliminating single points of failure (fault tolerance), and providing the ability to rapidly resume operations after a system outage with minimal business loss (fault resilience).

Fault recovery is the process of recovering and resuming operations after an outage due to a fault.

Disaster recovery is the process of recovering operations after an outage due to a prolonged data center or site failure. Preparing for disasters may require backing up data across longer distances, and may thus be more complex and costly.

High Availability: Power failure

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.

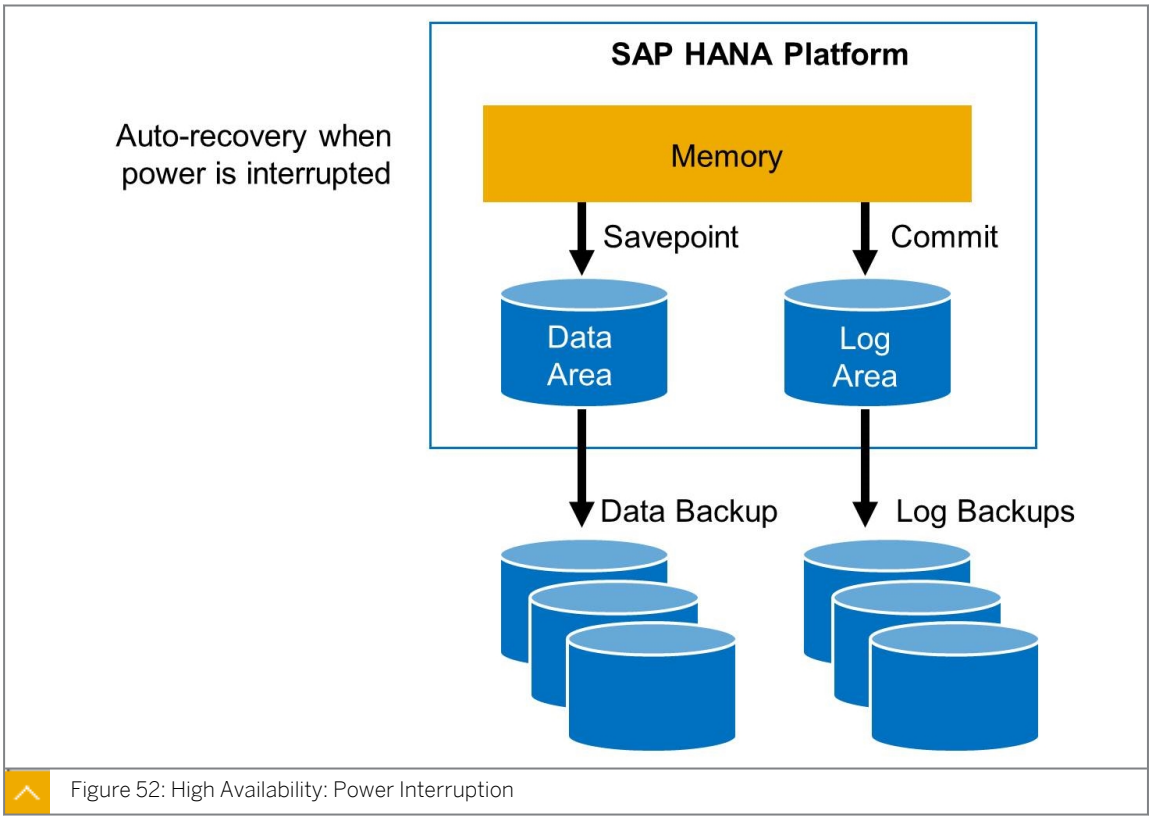


Figure 52: High Availability: Power Interruption

Every 5 minutes, SAP HANA automatically takes a delta data snapshot of the entire memory. It stores this delta information on the disk layer in an area called the data volume. This process 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.

However, it is important to have a mechanism to ensure that no data is lost, even between savepoints. To do this, every committed transaction is recorded and saved to the log volume. Thus, every update to the database since the last savepoint is captured.

When power is restored and the database system is started, the data is automatically reloaded into memory from the last savepoint and the open transactions are recovered by replaying the committed transactions. All committed transactions that can't be recovered because they weren't finished (committed) are rolled back to their original state.

This all happens automatically in the background.

Note:
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.

High Availability and Disaster Tolerance

Savepoints and transaction logging helps to prevent data loss during a power failure, but these measures aren't enough to survive hardware failures or natural disasters. To prepare for these circumstances, SAP HANA provides:

- Server failure: High Availability with Standby servers

- Data center failure: System Replication

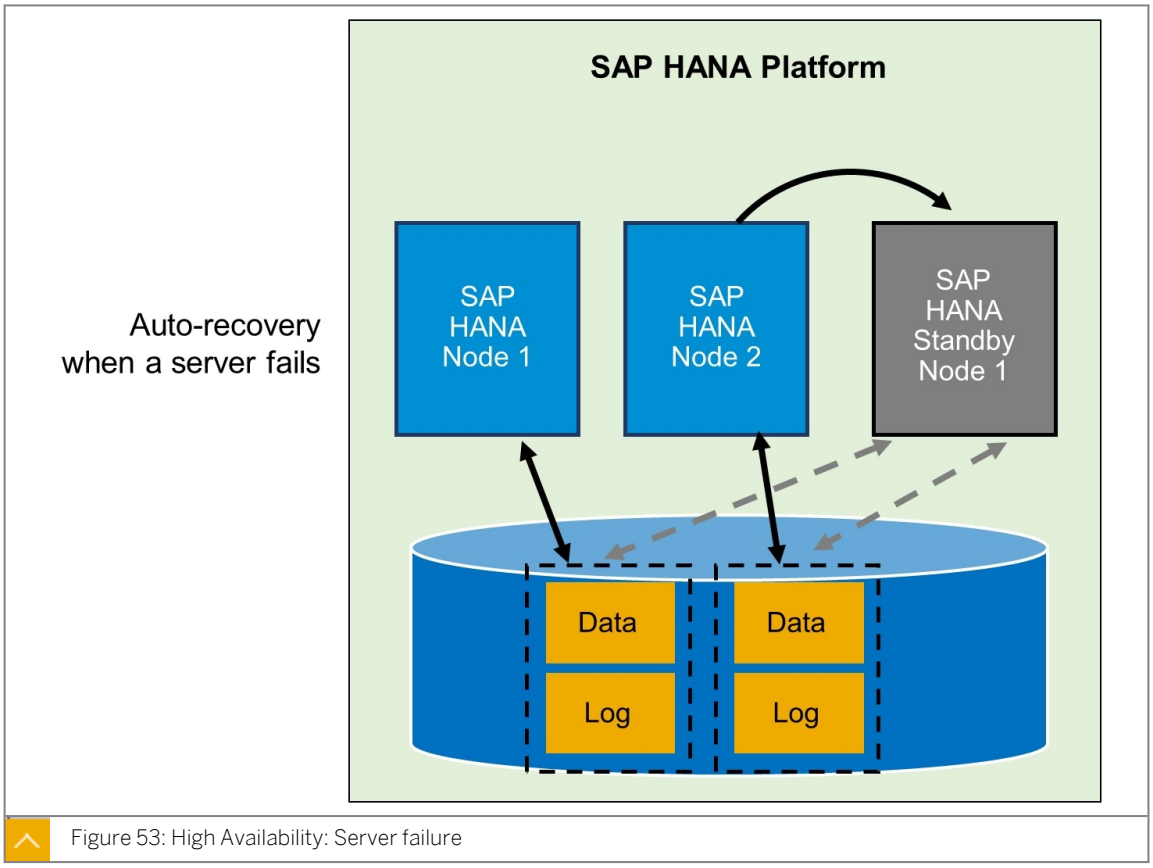


Figure 53: High Availability: Server failure

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 one or more 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.

A standby server is in a warm standby state, which means that it is only need to connect to the data- and log volume of the failed server. The standby server doesn't need to be started from a cold boot.

Caution:
A high available SAP HANA database system scale-out system with one or more standby server always resides in the same data center. Having the SAP HANA server spread between data center is NOT supported by SAP.

Disaster Tolerance: Data center failure

Standby servers can help avoiding system downtime is case of a server failure, but it can't prevent system downtime in case of a data center failure. SAP HANA provides SAP HANA system replication to keep SAP HANA up and running even if a data center is lost due to a natural disaster (fire, flooding, earthquake, ...) or a man-made disaster.

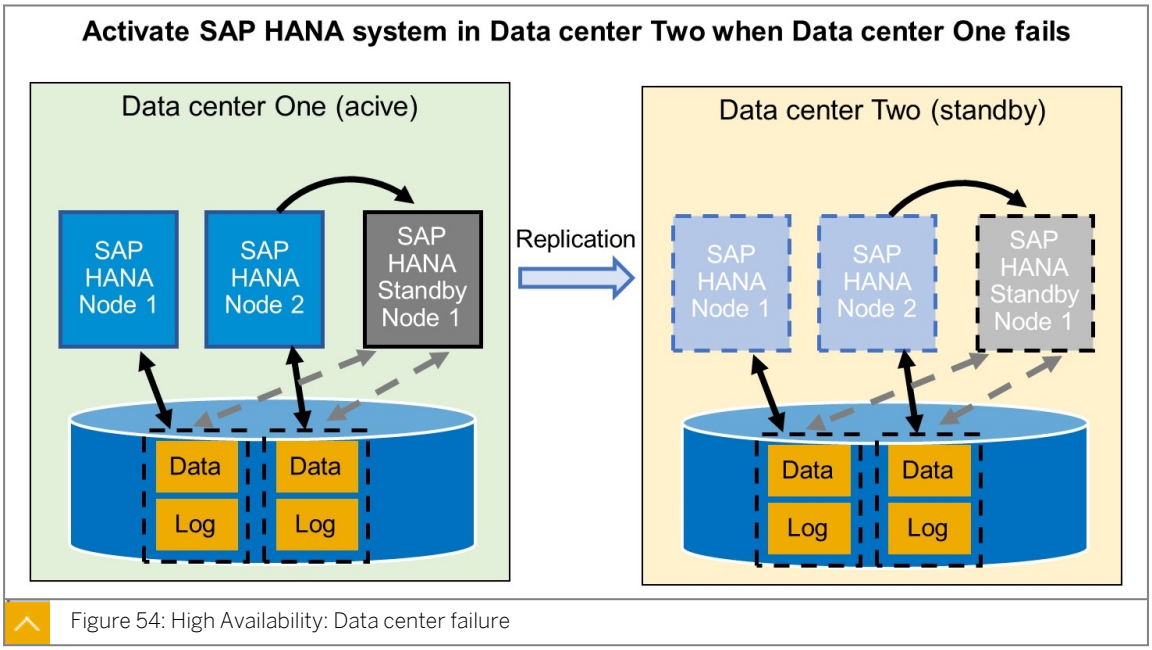


Figure 54: High Availability: Data center failure

SAP HANA system replication provides a technique were the primary system in database center One replicates the database in real time to a secondary system in data center Two. The SAP HANA system setup can be single host or scale-out.

The secondary system 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 system in data center Two. There is also no data loss because both systems are in-sync. This approach would be necessary for a mission-critical operation where downtime would be harmful to the business.

The data and log synchronization can be performed synchronous(short distances) and asynchronous (long distances). Also multiple synchronization layers (multi-tier) and multiple destinations are possible.

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.

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

Note:
For more information on all high-availability, refer to course HA201.

LESSON SUMMARY
You should now be able to:

- Describe high availability

Unit 2
Lesson 3

SAP HANA Lifecycle Management Tools

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

- Describe SAP HANA Lifecycle Management Tools
- Describe the SAP HANA Revision Strategy

The HDBLCM Tools

SAP HANA Lifecycle Management Tools

SAP HANA lifecycle management covers two aspects:

- Platform lifecycle management for installation, updating, configuring and customizing the SAP HANA database system software and components.
- Application lifecycle management for managing SAP HANA content products and transports for development, modeling and reports.

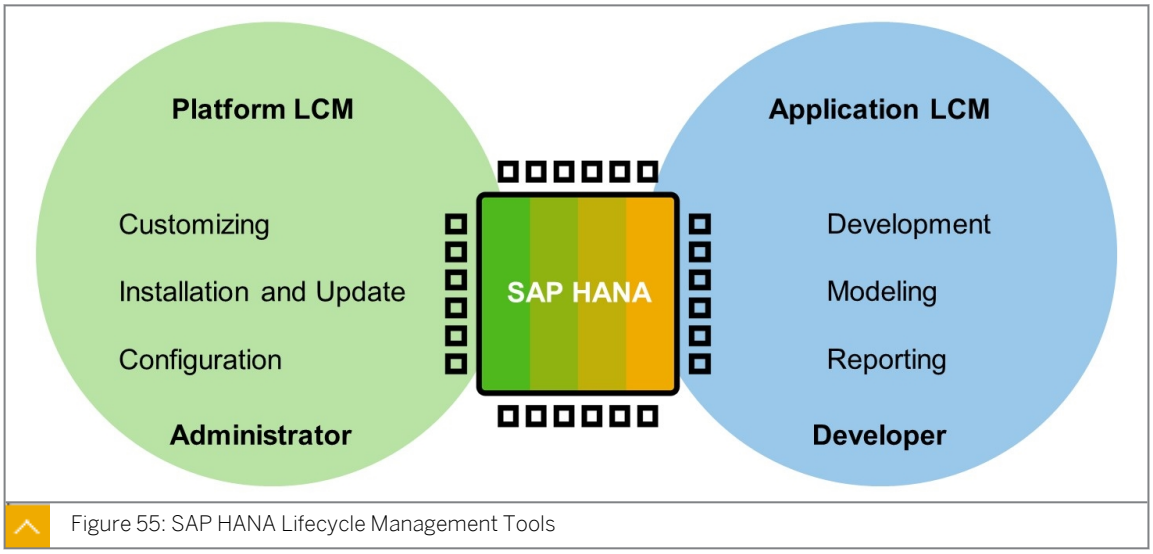


Figure 55: SAP HANA Lifecycle Management Tools

Platform Lifecycle Management Tools

You can customize platform lifecycle management aspects of your SAP HANA system by accessing the SAP HANA database lifecycle manager from three user interfaces: the graphical user interface, the command-line interface, or the Web user interface in a stand-alone Web browser, in the SAP HANA Cockpit.

SAP HANA platform lifecycle management encompasses the installation and update of an SAP HANA server, mandatory components, and additional components, as well as the post-installation configuration. The complete concepts and procedures for SAP HANA platform

installation and update are described in the SAP HANA Server Installation and Update Guide on the SAP Help Portal.

Several system configuration features are integrated into the SAP HANA database lifecycle manager, such as the following:

- The initial configuration of your SAP HANA platform to integrate it into your landscape. For example, by registering it in a system landscape directory, or configuring the inter-service communication.
- Adapting the topology of your SAP HANA platform by adding or removing additional SAP HANA hosts.
- Reconfiguring the system.

Application Lifecycle Management Tools

SAP HANA application lifecycle management tools can be accessed in different user interfaces: an interface that runs as an SAP HANA XS application in a web browser, a command-line tool hdbalm or via the SAP HANA cockpit.

SAP HANA application lifecycle management supports you in all phases of the lifecycle of an SAP HANA application or add-on product, from modelling your product structure, through application development, transport, assembly, to installing and updating products that you have downloaded from SAP Support Portal or which you have assembled yourself.

All application lifecycle management tasks are documented in the guide SAP HANA Application Lifecycle Management on the SAP Help Portal.

SAP HANA Database system administrators use SAP HANA application lifecycle management mainly to install and update SAP HANA applications or add-on products. Tasks related to SAP HANA development are documented in the SAP HANA Developer Guide - For SAP HANA Web Workbench on the SAP Help Portal.

The HDBLCM User Interfaces

The SAP HANA platform lifecycle management tool can be used to install and update an SAP HANA database system and to add mandatory components and additional components.

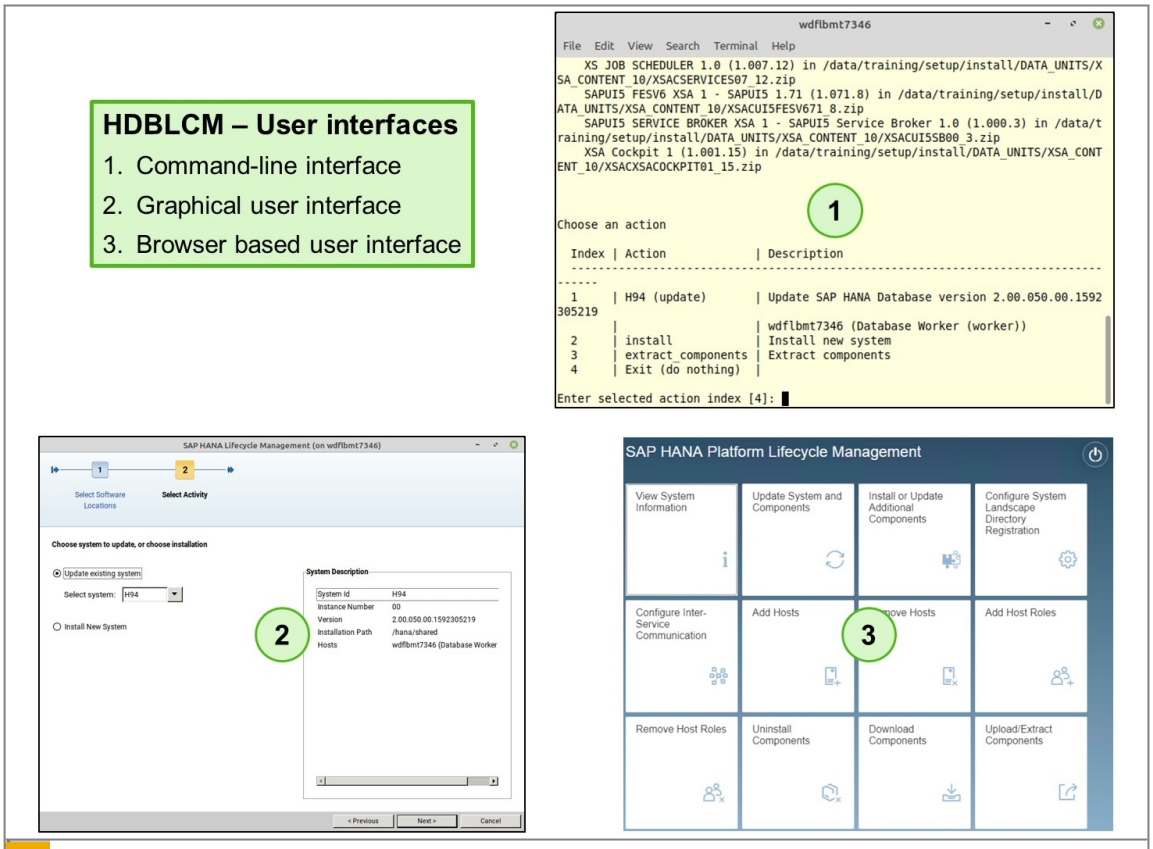


Figure 56: The HDBLCM User Interfaces

The SAP HANA database lifecycle management tools also provides a browser-based user interface, but this can only be used for SAP HANA updates and several other lifecycle management tasks.

The SAP HANA database system can be installed using the command-line user interface or by using a graphical user interface. For advanced Tailored Data Center integration (TDI) installations, the command-line interface with a configuration file is the best option.

SAP HANA Maintenance Strategy

The SAP HANA maintenance strategy is based on incremental, non-disruptive innovation updates.

New functions are introduced once a year, every time a new SAP HANA Support Package Stack (SPS) is released. This happens normally in the month of April.

As of SAP HANA 2.0 SPS02, SAP provides bug fixes and security patches for every SPS for two years after Release To Customer (RTC). The last SPS of a major product version is in maintenance for five years after RTC.

Because updates shipped for the SAP HANA platform are strictly downward compatible, earlier revisions can be removed from SAP Support Portal when a newer SAP HANA revision of the same SPS becomes available. Incompatible changes are considered for legal or security reasons, but are subject to a strict exception approval process.

Note:
For more information, see SAP Note 2378962 - SAP HANA 2.0 Revision and Maintenance Strategy.

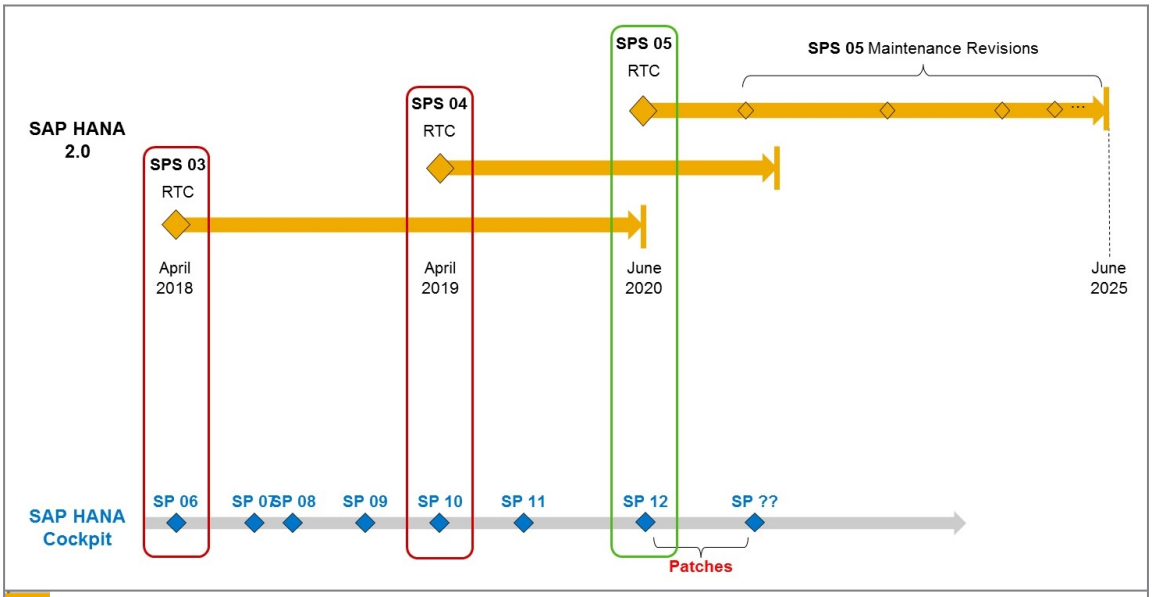


Figure 57: SAP HANA and SAP HANA Cockpit Revision Strategy

SAP HANA Cockpit 2.0 Revision Strategy

A SAP HANA 2.0 cockpit installation which comprises of XS Advanced, a set of XSA multi-target applications, and SAP HANA, express edition, receive new functionality, updates, patches and corrections on a regular basis.

For the SAP HANA Platform, the Support Package Stacks are released on a yearly basis. These Support Package Stacks introduce new capabilities into SAP HANA Platform. A new SAP HANA cockpit is released as part of the SAP HANA Platform Support Package Stack. In between that yearly release cycle, SAP HANA 2.0 cockpit support packages can be released.

Note:
For more information, see SAP Note 2433181 - SAP HANA 2.0 Cockpit Revision and Maintenance Strategy

LESSON SUMMARY

You should now be able to:

- Describe SAP HANA Lifecycle Management Tools
- Describe the SAP HANA Revision Strategy

Learning Assessment

1. What methods can you use to size a SAP HANA database system?

   Choose the correct answers.

   - ☐ A Use the SAP Quick Sizer website
   - ☐ B Use the The SAP HANA Hardware Directory website
   - ☐ C Use the ABAP report RSPFPAR
   - ☐ D Use the ABAP report /SDF/HDB_SIZING

2. What are scale-out scenarios with SAP HANA?

   Choose the correct answers.

   - ☐ A Provision of extra redundant standby servers that can be switched to in case of hardware failure.
   - ☐ B Deployment of low-cost generic servers to support high volume data streaming applications.
   - ☐ C Use of multiple servers to increase data storage capacity.

3. Which features are supported by the SAP HANA platform lifecycle management tools?

   Choose the correct answers.

   - ☐ A Adding or removing additional SAP HANA hosts.
   - ☐ B Creating and deploying SAP HANA CDS views.
   - ☐ C Installing or updating an SAP HANA database system.
   - ☐ D Performing SAP HANA database backups.

4. What is the correct SAP HANA Maintenance Strategy?

   Choose the correct answer.

   - ☐ A A new release every 6 months and 3 months maintenance support.
   - ☐ B A new release every year and 2 years maintenance support.
   - ☐ C A new release every 6 months and 2 years maintenance support.

Learning Assessment - Answers

1. What methods can you use to size a SAP HANA database system?

   Choose the correct answers.

   - [X] A Use the SAP Quick Sizer website
   - [ ] B Use the The SAP HANA Hardware Directory website
   - [ ] C Use the ABAP report RSPFPAR
   - [X] D Use the ABAP report /SDF/HDB_SIZING

   Correct, SAP HANA sizing can be performed using the SAP Quick Sizer website and/or ABAP report /SDF/HDB_SIZING.

2. What are scale-out scenarios with SAP HANA?

   Choose the correct answers.

   - [X] A Provision of extra redundant standby servers that can be switched to in case of hardware failure.
   - [ ] B Deployment of low-cost generic servers to support high volume data streaming applications.
   - [X] C Use of multiple servers to increase data storage capacity.

   Correct — Scale-out is the use of redundant standby servers that we can switch over to in case of primary server failure. It is also the use of multiple servers to increase data storage capacity.

3. Which features are supported by the SAP HANA platform lifecycle management tools?

   Choose the correct answers.

   - [X] A Adding or removing additional SAP HANA hosts.
   - [ ] B Creating and deploying SAP HANA CDS views.
   - [X] C Installing or updating an SAP HANA database system.
   - [ ] D Performing SAP HANA database backups.

   Correct, the SAP HANA platform lifecycle management tools can be used for adding or removing SAP HANA hosts and to install or update a SAP HANA system.

4. What is the correct SAP HANA Maintenance Strategy?

   Choose the correct answer.

   - [ ] A A new release every 6 months and 3 months maintenance support.
   - [x] B A new release every year and 2 years maintenance support.
   - [ ] C A new release every 6 months and 2 years maintenance support.

   Indeed. SAP HANA gets a new release every year, and this new release gets 2 years maintenance support.

UNIT 3 Analytical Processing with SAP HANA

UNIT OBJECTIVES

- Develop SAP HANA calculation views
- Advanced Analytics with SAP HANA
- Connect SAP Business Intelligence Tools to SAP HANA

Unit 3
Lesson 1

Developing Calculation Views 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:

- Develop SAP HANA calculation views

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.

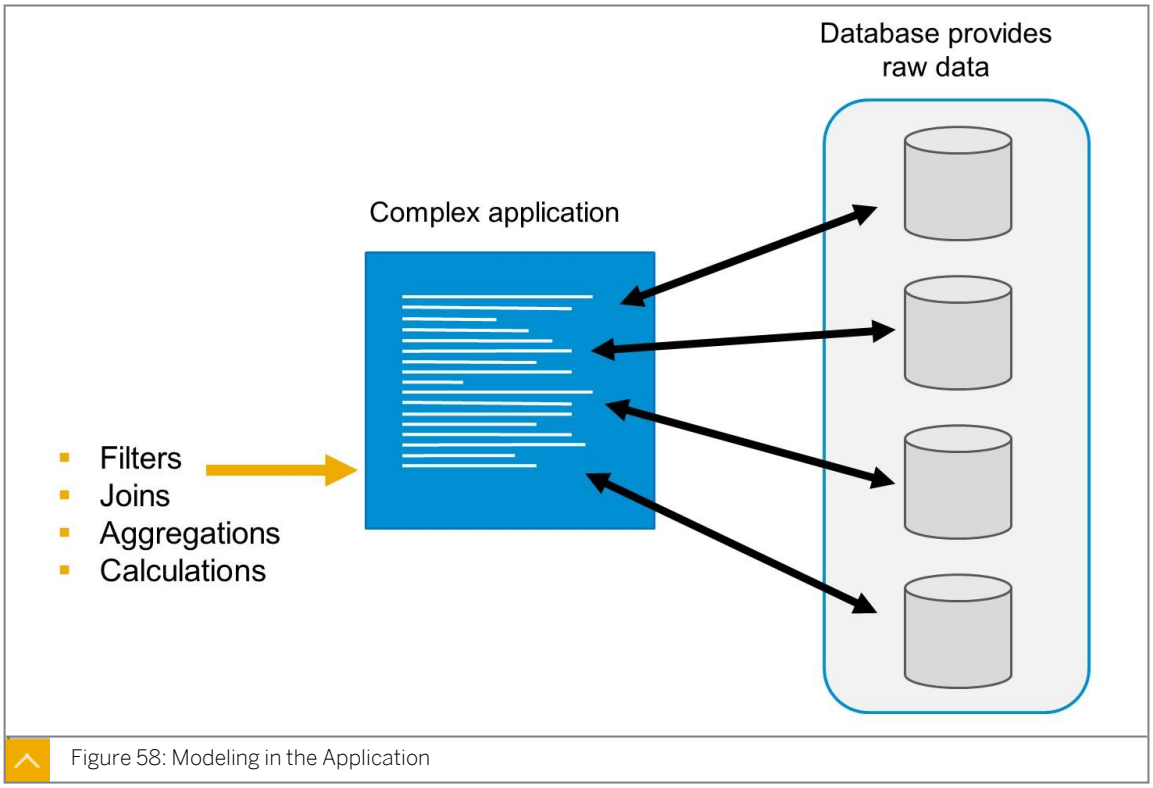


Figure 58: Modeling in the Application

In a traditional application, the role of the database is to 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.

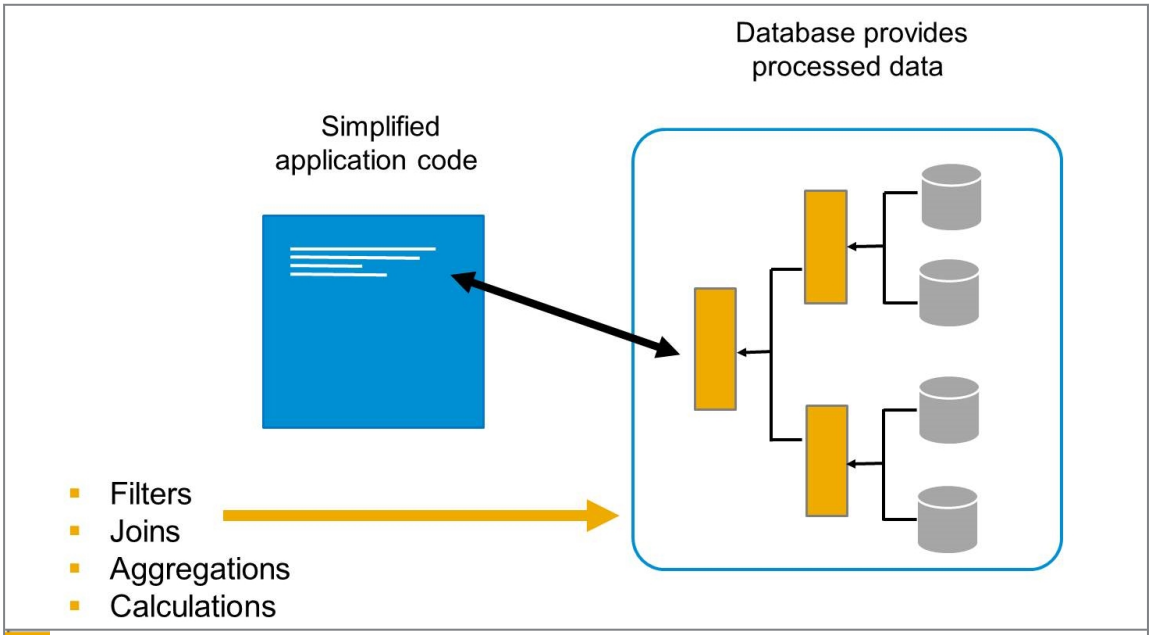


Figure 59: Modeling in the Database

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.

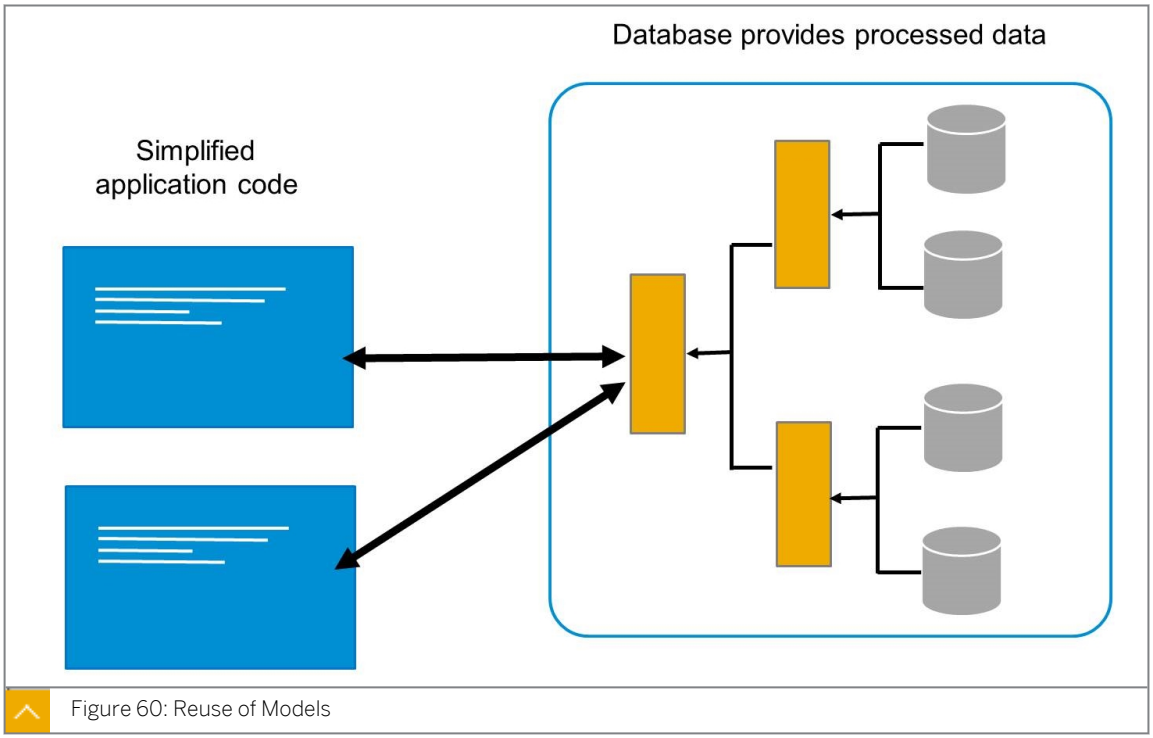


Figure 60: Reuse of Models

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 view 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.

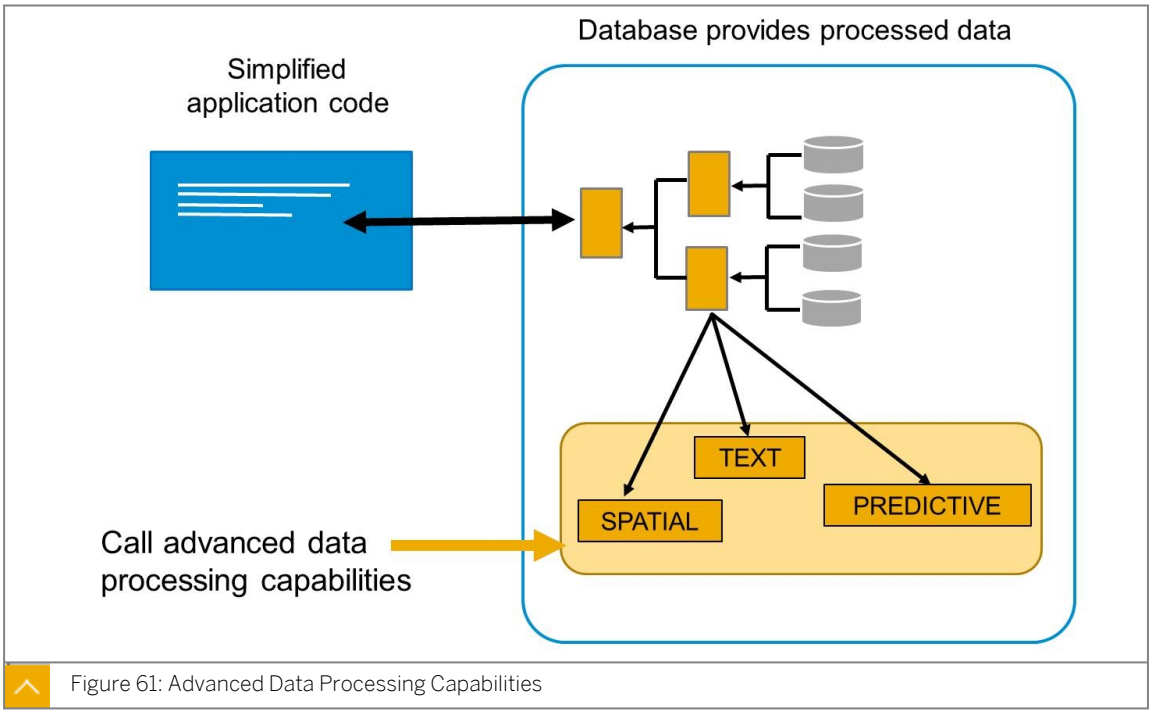


Figure 61: Advanced Data Processing Capabilities

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.

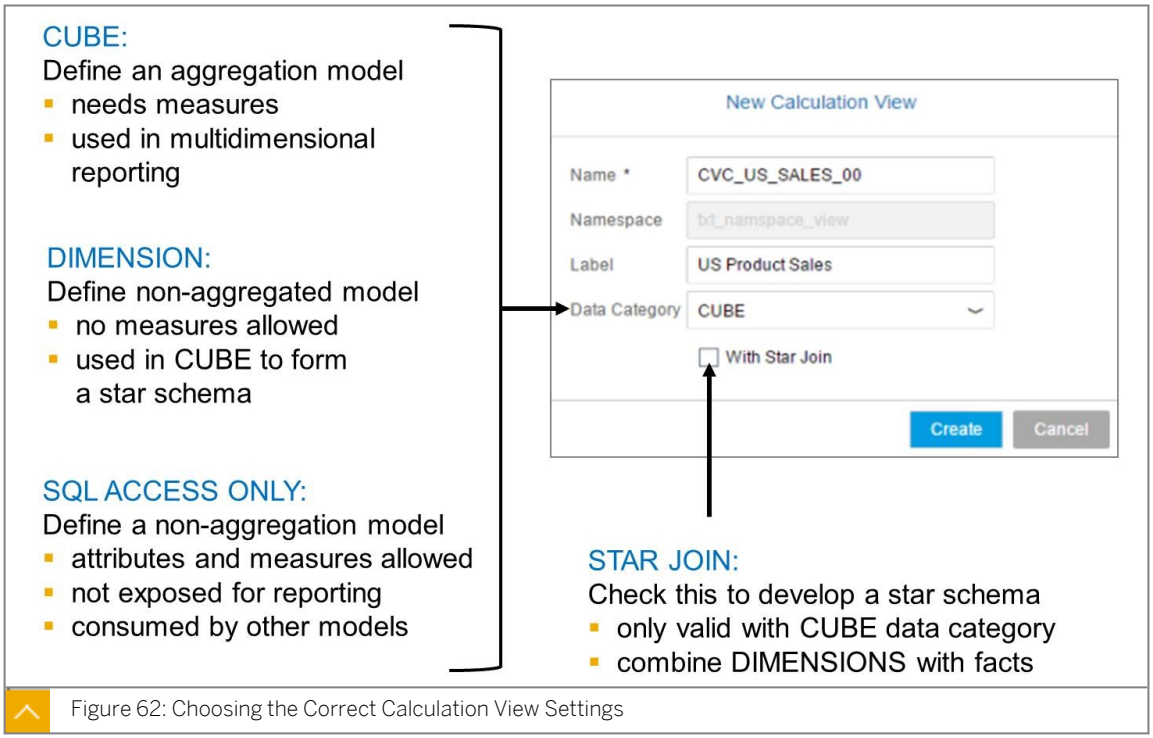


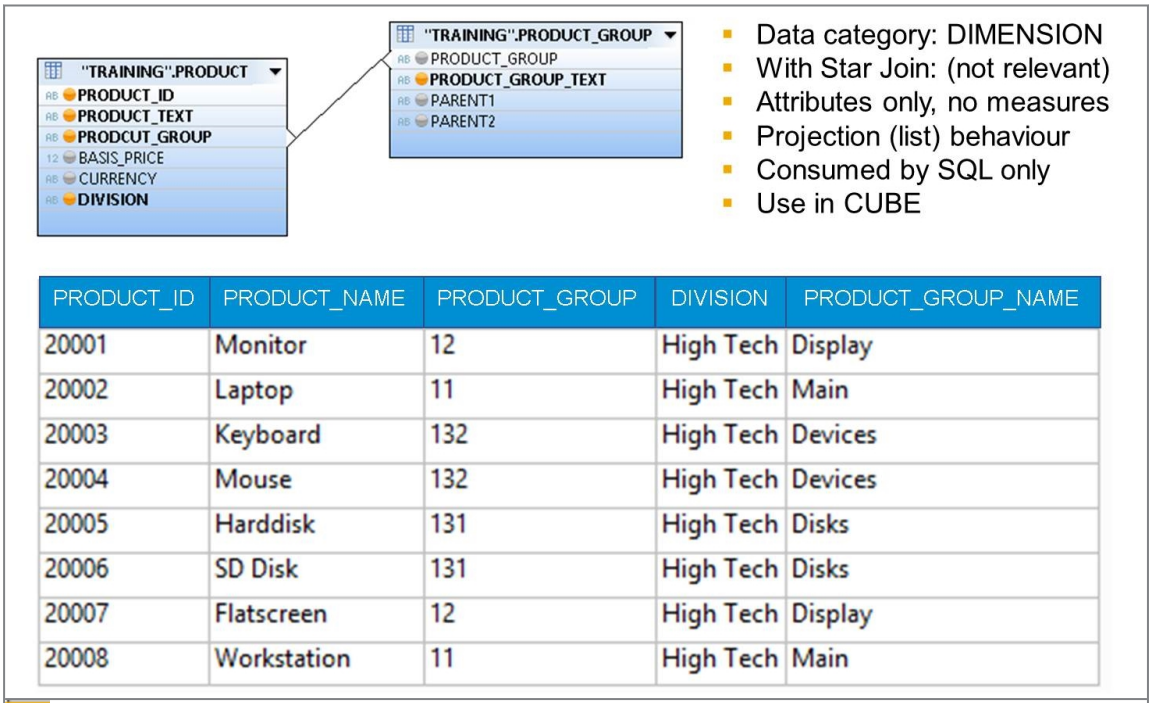
Figure 62: Choosing the Correct Calculation View Settings

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

- Dimension
- Cube without star schema
- Cube with star schema
- SQL Access Only

Modeling Dimensions

The purpose of a dimension type of calculation view is to define a group of related attributes, such as material, material color, material weight, and material price. Although this type of calculation view can be directly consumed by SQL, it is most likely to be found as a reusable component to support a calculation view of the type *Cube with star schema*.



PRODUCT_ID	PRODUCT_NAME	PRODUCT_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 63: Modeling Dimensions

Dimension type calculation views cannot contain measures, they can only contain attributes. This means that any numerical value such as *days vacation remaining* can only be defined as attribute in this type of calculation view. This is still useful to provide supporting information about a main entity, in this case an employee. The value cannot be defined as a measure as they are not allowed so you could not define an aggregation behavior such as sum, average and so on.

Reporting tools usually cannot directly access calculation views of type *DIMENSION*.

It might be helpful to think of calculation views of type dimension as **central master data** views. You define them once and reuse them many times.

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, joins if needed, filters, and identify the output 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.

In the calculation view, you are able to rename any source columns to be meaningful to the business user. Remember that the column names originate from the database tables, and these names can often be meaningless to business users.

Modeling Cubes

The next type of calculation view is the type *Cube*. This type of calculation view is used to define a data set comprised of attributes and measures. This means they are ideal for transactional data modeling. With measures you are able to define aggregation behavior so that business users can slide and dice the data in an ad-hoc manner. It should be noted that a *cube* calculation view is not a star schema as there are no dimensions defined.

- Data category: CUBE
- With Star Join: No
- Needs measures
- Aggregation behaviour
- Consumed directly by reports and SQL

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 64: Modeling Cubes

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

To get started, set the data category to *CUBE*.

Do not select the *Star Join* flag.

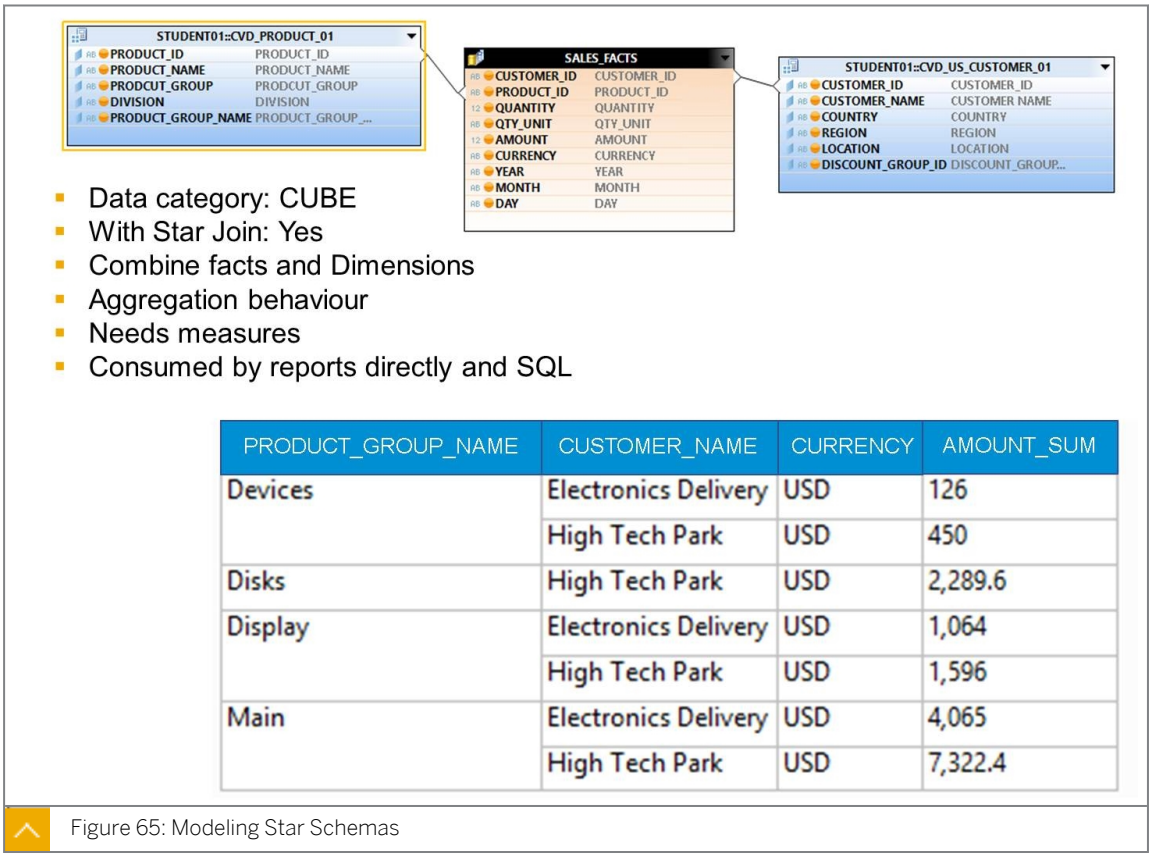
You then select the table, or tables, which are to be included in the model. Typically, you choose a transaction table so that you have columns from which you can define attributes and measures. It is possible to include more than one table. For example, you may need to include a header and a line item 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 business user.

Modeling Star Schemas

Now we come to the most powerful of all the calculation views: the *Cube* type, *but* with an additional setting — **star join**. This is how we model a star schema. This is based on the *cube* type of calculation view, but with one or more dimension calculation views joined to the model. This provides a very powerful and flexible data model that enables drill-down, slice and dice analysis over any combination of attributes and measures. Adding dimensions dramatically enhances the analysis possibilities over simply *cube* type calculation views.



PRODUCT_GROUP_NAME	CUSTOMER_NAME	CURRENCY	AMOUNT_SUM
Devices	Electronics Delivery	USD	126
	High Tech Park	USD	450
Disks	High Tech Park	USD	2,289.6
Display	Electronics Delivery	USD	1,064
	High Tech Park	USD	1,596
Main	Electronics Delivery	USD	4,065
	High Tech Park	USD	7,322.4

Figure 65: Modeling Star Schemas

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

First, select the transaction table(s) and create joins to combine the transaction tables if required. 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 is used as the hub of the star schema.

The next step is to form the star schema by linking suitable calculation views of type *DIMENSION* to the fact table.

Finally, improve the names of any columns by using the rename function in the *Semantic* node so that it becomes fully business user facing.

Modeling Internal Views

Calculation views are usually modeled on top of each other in a stack with each calculation view consuming lower level calculation views. Only the lower calculation views usually consume tables. The very top calculation view is the only view that needs to be visible to the consuming application or reporting tool.

SAP has provided a general purpose calculation view that has the data category *SQL ACCESS ONLY*. 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 a star schema and cannot contain dimensions. It is not visible to reporting tools and is consumed only by SQL or by other calculation views. Unlike other calculation views, this view does not expose its metadata to the consuming applications. It might help to think of this type of view as an *internal view* that is only meant to be used within a stack.

Note:
This type of calculation view was previously known as *DEFAULT* but since SAP HANA 2.0 SPS04, the name was changed to provide something more meaningful to developers. The functionality did not change.

SAP HANA Modeling Editor

Calculation views are created using the *Development View* of the Web IDE for SAP HANA.

With Web IDE you can create the following modeling artifacts:

- Calculation View
- Procedure
- Function
- Flowgraph
- Analytic Privilege
- CDS artifacts
- Virtual Table
- Replication Task

The objects are created using either graphical editors or code based editors. Even for artifacts created using the graphical editor, it is usually possible to continue to develop using the code based editor and vice-versa.

The syntax for some artifacts is complex, such as for calculation views, so the graphical editor should be used.

Regardless of whether you use graphical editors or code based editors, the artifact definition is stored in a simple text-based source file which can easily be copy / pasted and generally shared with others (for example, email) , just like a simple *.TXT* file. You can even open the file in a text editor outside Web IDE and view the source code. This makes debugging simple when you can pass the file around easily.

For version control, the open-source Git framework is used. Git is fully integrated in Web IDE. Git allows developers to clone a project and work in parallel with other developers on sections of the project without affecting others. When development is complete the artifacts can be merged together using Git.

Extending Calculation Views with SQLScript

Calculation views are built using a graphical approach and no SQL 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). In this section we focus on table functions.

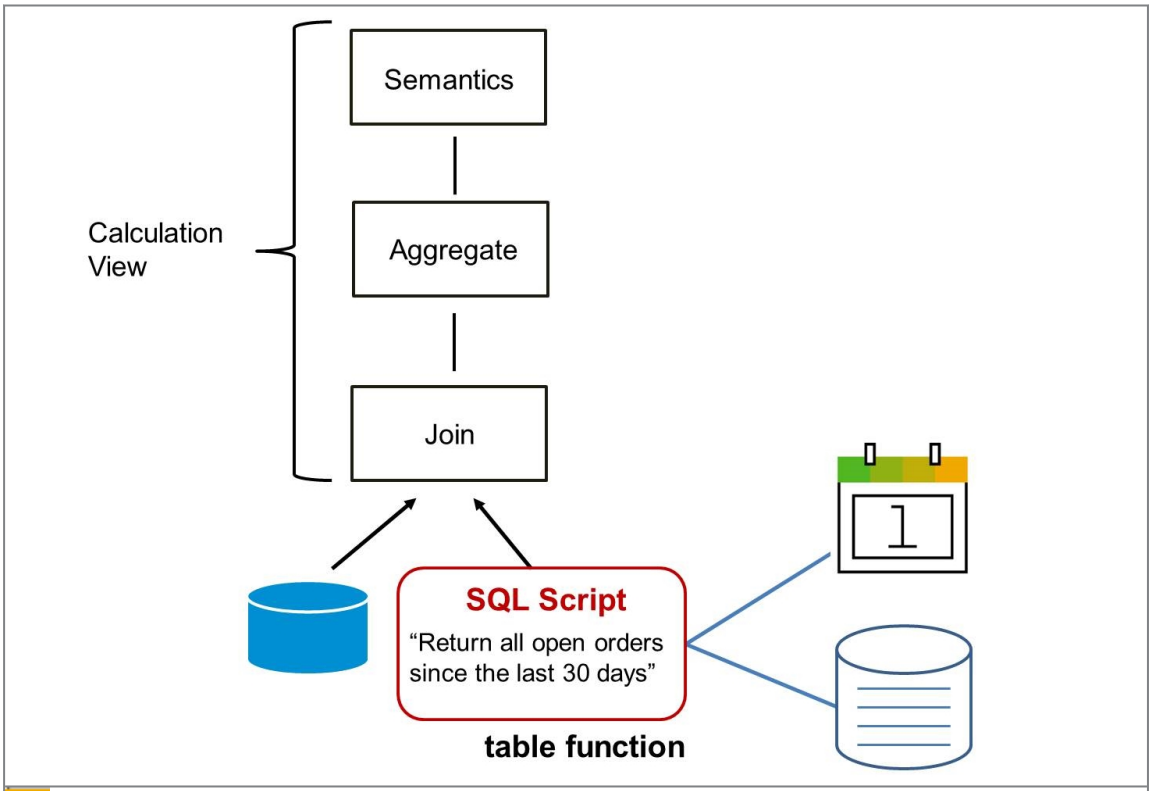


Figure 66: Define Source of Data Using Table Functions

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 identifier 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.

Add SQL to Models Using 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 SQLScript 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 structure. Procedures have a limited role in data modeling. Procedures cannot be used as data sources to calculation views but they are used in other places in a calculation view, for example, to automatically derive values for an input parameter.

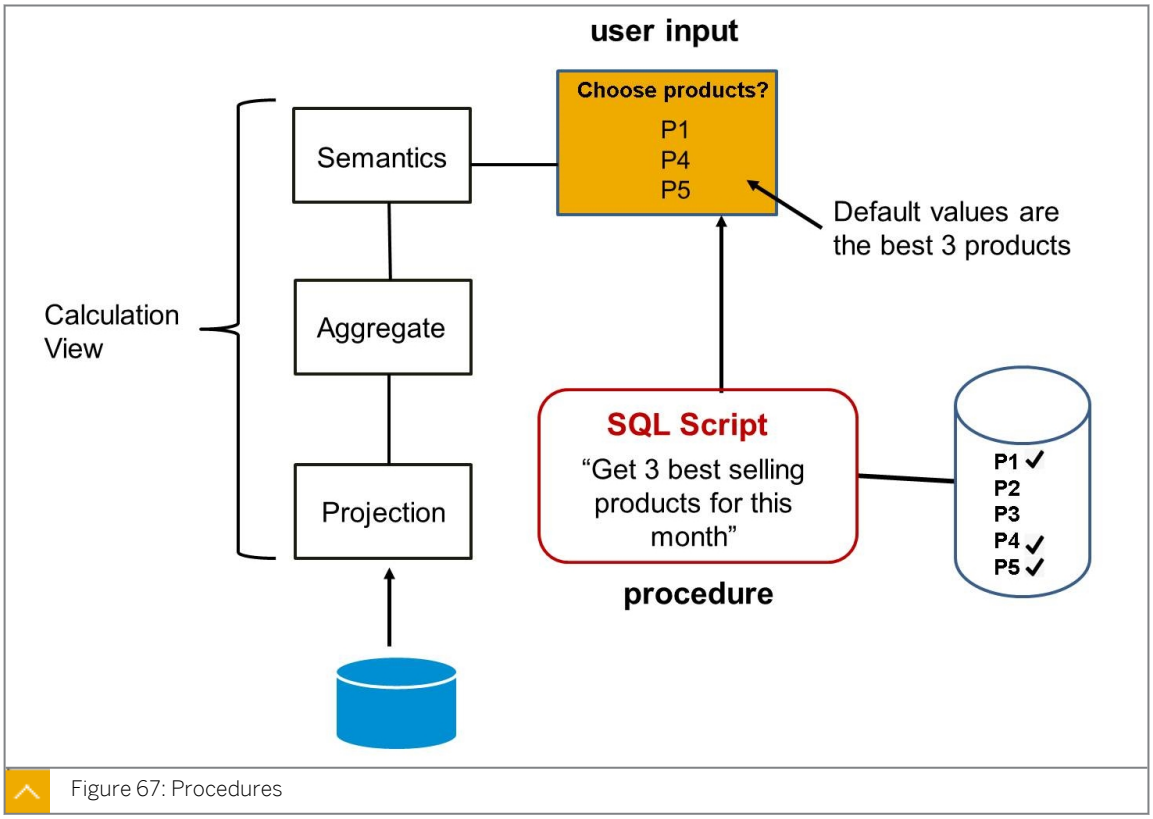


Figure 67: Procedures

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 must be read only. In that case they are called stateless (or side-effect free), as they don't alter any data or meta data in the database. However, procedures can also be used to update, insert, and delete data if required. These procedures are called stateful, and these type are not allowed when called from calculation views. Stateful procedures are more likely to be used by developers who build applications rather than modelers.

LESSON SUMMARY

You should now be able to:

- Develop SAP HANA calculation views

Unit 3
Lesson 2

Advanced Analytics with SAP HANA

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

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

- Advanced Analytics with SAP HANA

Text Analytics with SAP HANA

Introducing SAP HANA Text Analytics

The vast majority of digital data is unstructured and a significant portion of that unstructured data is textual data. SAP HANA provides powerful text processing capabilities that can generate high value results from text. SAP HANA Text Analytics can be performed over many languages (32 at the current time) and on many text friendly data types. We can process text from many types of documents including PDF, PPT, DOC, HTML, XML, Outlook. This also includes binary documents.

SAP HANA Text Analytics can be broken up in to three major analytic capabilities:

- Full text search
- Text Analysis
- Text Mining

Full Text Search

Let's take a brief look at text search. It is called Full Text Search as opposed to simply Text Search because most text searches operate on individuals words. But SAP HANA Full Text Search works across entire phrases. This provides very powerful search possibilities.

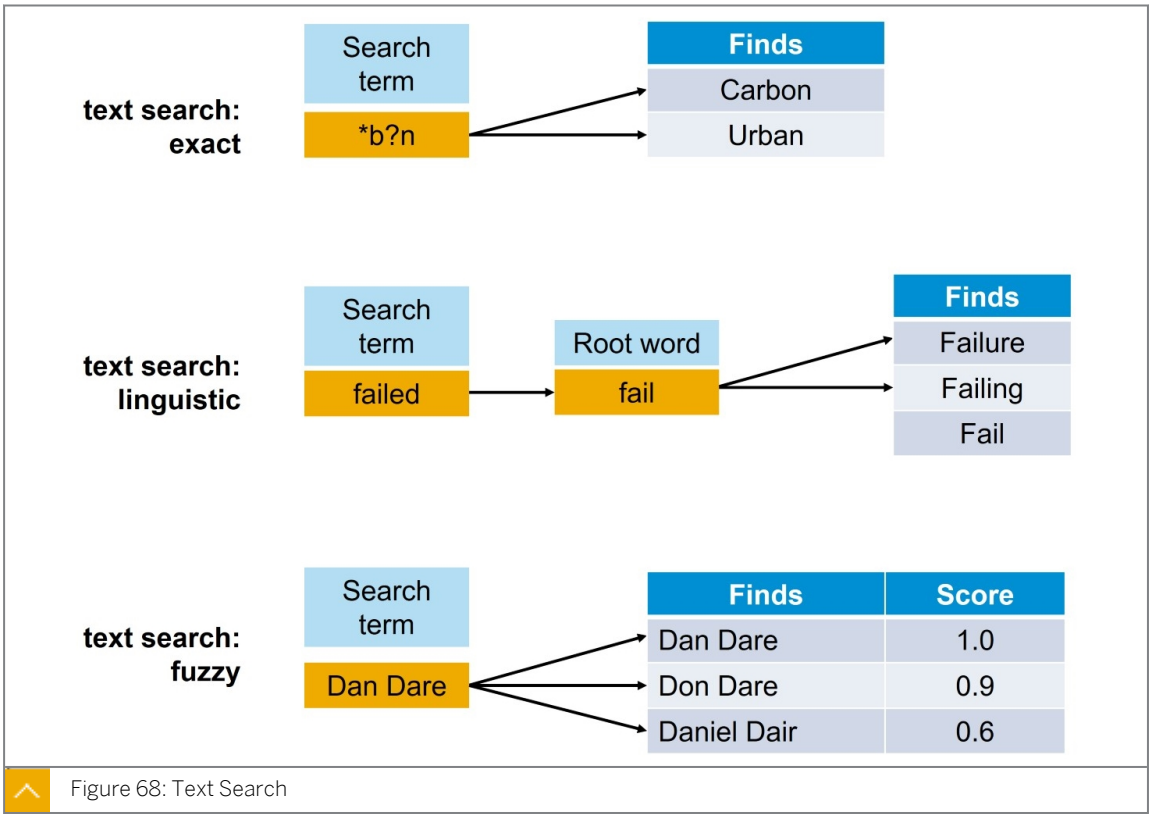


Figure 68: Text Search

With SAP HANA Full Text Search you can perform different types of text search:

- Exact search
  - — Search using wild cards pattern matching (for example,'*bo?' finds 'labor' and 'robot')
- Linguistic search
  - — Search using a linguistic analysis (for example, 'speaking' finds 'speak' and 'spoken')
- Fuzzy search
  - — Fault-tolerant search (for example, 'ACME' finds 'AMCE')

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. Once the linguistics has been extracted from the text, we can perform powerful analytics on the results. For example, we can identify the subject, urgency, sentiment, and more.

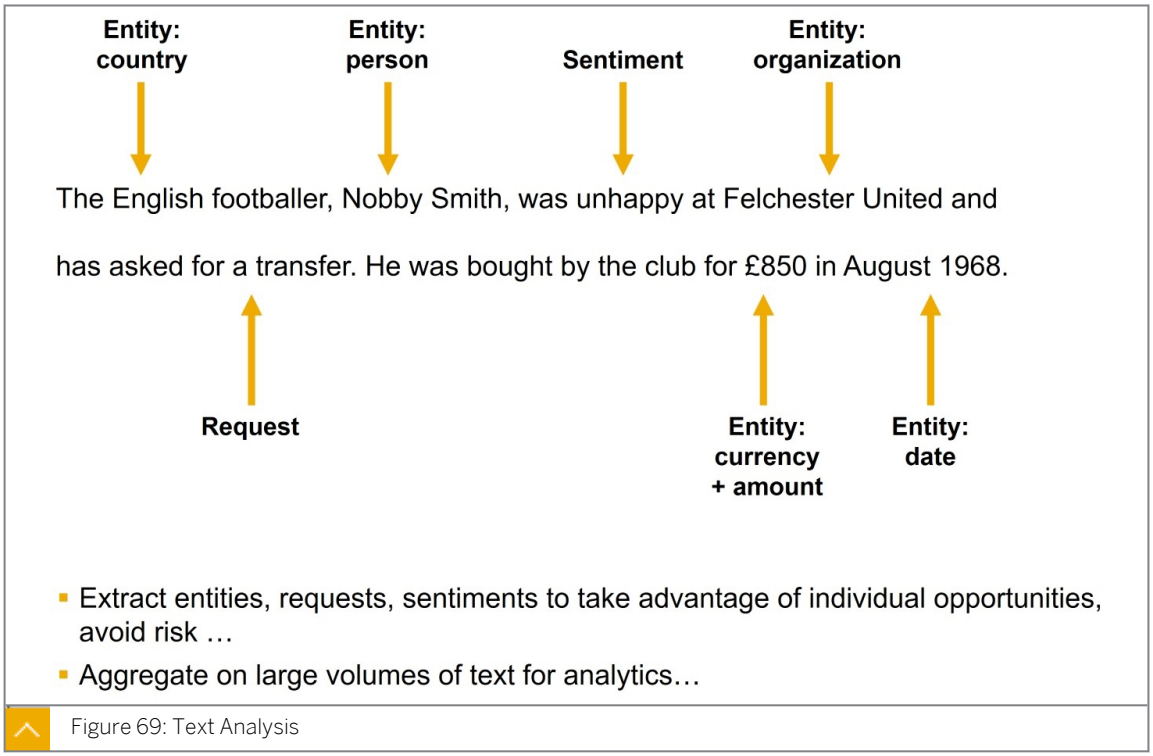


Figure 69: Text Analysis

The results of a Text Analysis are stored in a table and can therefore be easily consumed through all supported SAP HANA scenarios.

SAP provides ready-to-go reference dictionaries of entities in multiple languages that can help identify interesting words that refer to organizations, dates, countries, and so on. SAP also supplies a populated 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 easily create their own custom dictionaries to include words and expressions used in their organizations.

Text Mining

In contrast to text analysis, which operates on the words inside a document, text mining operates at the document level. In other words, text mining is used when the whole document is of interest and not just an analysis of some words or phrases within it.

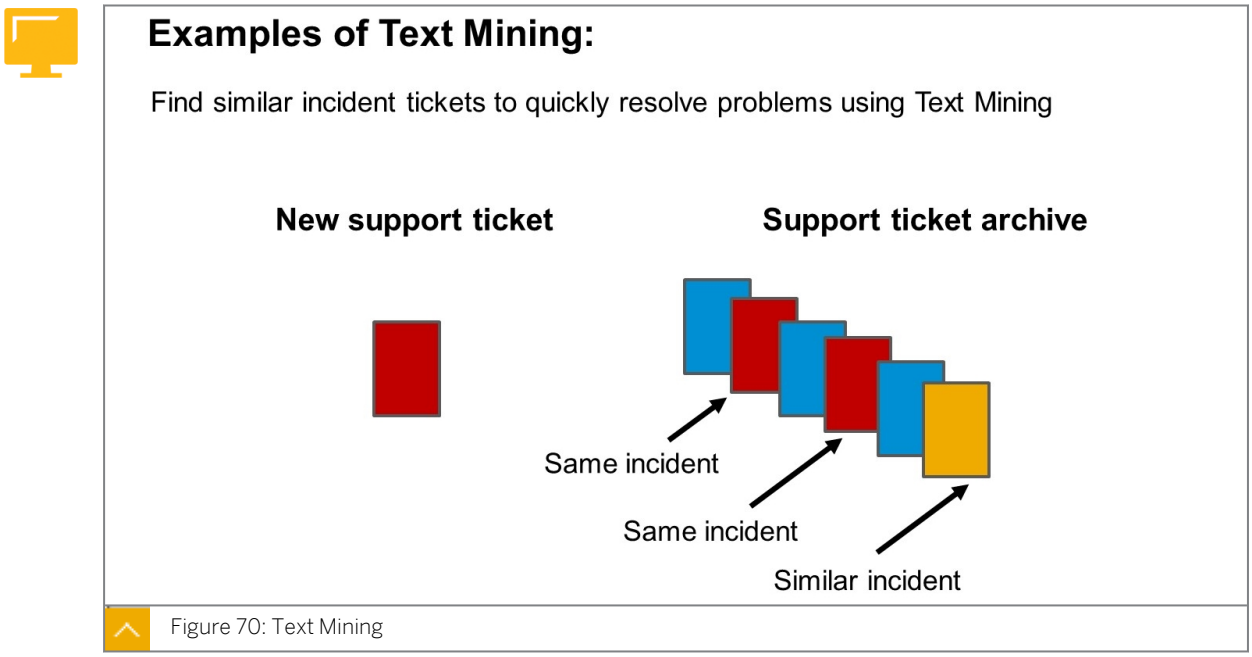


Figure 70: Text Mining

Text mining works with groups of documents to compare and classify using statistical analysis and not linguistic analysis. A key use case for text mining is to automatically classify streams of documents, such as newspaper articles, blogs, or support incidents. Or to locate similar documents, such as medical journal that discuss common topics. Each document is preprocessed and the strong subjects that they contain are noted and can be used to compare with other documents.

SAP HANA Spatial

By combining traditional business data with spatial data, you can build innovative applications that can provide deeper insight to unlock opportunities or reduce risk.

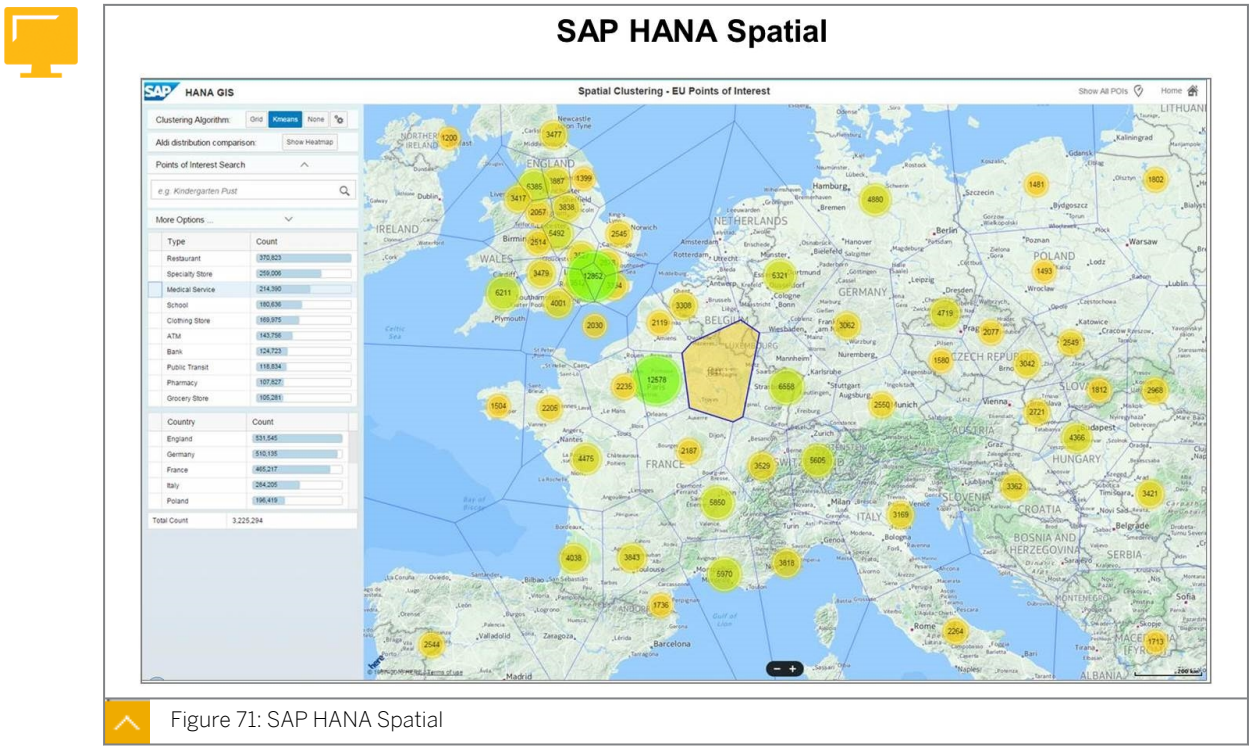


Figure 71: SAP HANA Spatial

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 identify 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 quickly. 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 more exciting use cases for spatial analysis in the sports environment. SAP has 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.

SAP Spatial provides new data types for storing geometrical data such as points, lines, and polygons. These can be used to represent precise locations, roads, and regions. SAP HANA Spatial uses open standards and so can easily be integrated with well-known, leading geospatial providers such as ESRI, OGC, OpenStreepMap, GoogleMap.

As well as storing spatial data, SAP HANA also provides spatial query functions that can easily be included in SQL Script. Here are some examples of the functions:

- Within — which customers are in my region?
- Distance — what is the longest distance a high-value customer has to travel to reach my sales outlet?
- Crosses — where does truck route A cross truck route B?

SAP HANA Spatial also provides algorithms that can determine clusters. This helps an organization to locate precise locations that might be lucrative based on income data and other interesting attributes associated with consumers.

Predictive Analysis

With SAP HANA, you can develop predictive models using a variety of approaches. These approaches include using:

- Predictive Analysis Library (PAL)
- Automated Predictive Library (APL)
- Extended Machine Learning (EML)
- R (algorithm library)

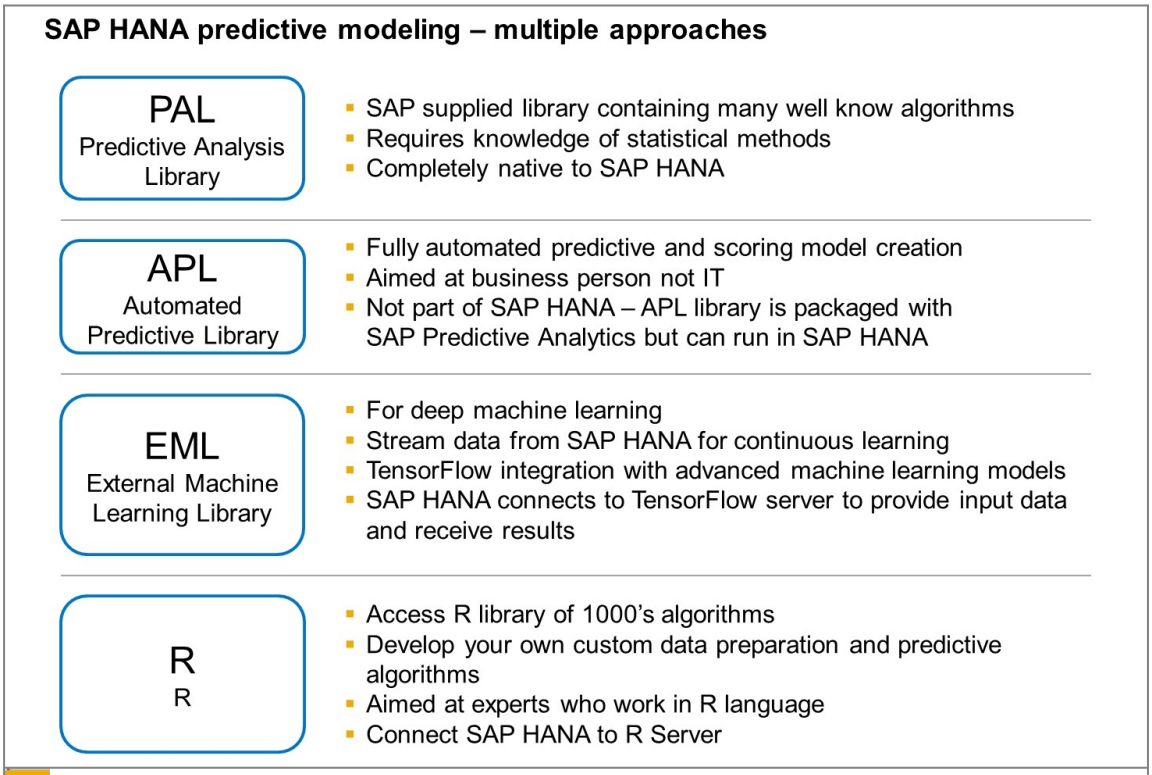


Figure 72: Predictive Analysis Approaches

SAP HANA Predictive Analysis Library (PAL)

SAP HANA Predictive Analysis Library (PAL) contains over 90 algorithms that can be used to develop predictive and machine learning models. Some of these algorithms are used for data mining pre-processing tasks such as:

- Sampling — select a few records from large data sets (for example, we need 1000 people from each country).
- Binnig — grouping records into basic categories (for example, age ranges).
- Partitioning — creating sets of data for training, testing, and validation used to train models and check their predictive accuracy.

The majority of the algorithms are used for scoring or predictive modeling. There are many algorithms provided for all major data mining categories including:

- Association
- Classification
- Clustering
- Regression
- Time Series
- Neural Networks

PAL algorithms can be called directly from procedures in SQL Script or they can be integrated into an SAP HANA flowgraph which is built using a graphical editor in the Web IDE. A flowgraph defines the data inputs, data processing, and outputs and parameters used in the predictive model. Using PAL requires knowledge of statistical methods and data mining

techniques. This is because the choice of which algorithm to use must be made by the developer. So it is important that the developer initially understands the differences between the algorithms used for data preparation, scoring, and prediction. But they must also know how to fine-tune the algorithms to reach to desired outcome. For example, a developer would need to decide when to use the time series algorithm for double exponential smoothing versus triple exponential smoothing and then how to adjust the parameters to consider trend or to eliminate outliers. Developers who work with SAP HANA PAL are typically already working in predictive analysis projects or have a reasonable understanding of the topic.

Automated Predictive Library (APL)

The Automated Predictive Library (APL) is not shipped with SAP HANA but belongs to the product SAP Predictive Analytics. For customers who use this product they can run their models in SAP HANA and that is why it is mentioned here. APL is aimed at business users who do not have (or desire to have) detailed knowledge of the algorithms and the maths behind the models. The selection of algorithms for data preparations, scoring, and predictions is completely automated (hence the name). All the business user has to do is to provide the data and APL finds the best data preparation and predictive models.

External Machine Learning Library (EML)

SAP HANA provides access to external libraries of machine learning models. Google TensorFlow was the first library to integrate with SAP HANA and today provides access to a large library of deep machine learning models. Examples of deep machine learning could include voice recognition, handwriting recognition, image recognition, and more. The machine learning models are created in the TensorFlow framework and a TensorFlow client is installed in SAP HANA to create the connection to the TensorFlow server. SAP HANA sends the source data, such as an image, to the TensorFlow server where it is processed (either in training mode, testing, or production mode) and the result is passed back to SAP HANA.

Google Tensorflow has become a very popular and a vastly growing framework open sourced by Google for machine learning data flows and its recent integration with SAP HANA is essential for the support of the growing applications for IoT.

R

R is an open source programming language used to develop statistical models. The R library contains thousands of freely accessible algorithms that can be used or adapted to provide custom predictive and data preparation capabilities. SAP HANA connects to an R server where the data is processed and the results are passed back to SAP HANA. R language is becoming increasingly popular among Data Scientists as the standardized language for statistical computing.

Graph Modeling

Graphs are used to model data that is best represented using a network. Examples include supply chain networks, transportation networks, 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 with lines (edges) to create a network. This network represents how each node relates to all other nodes.

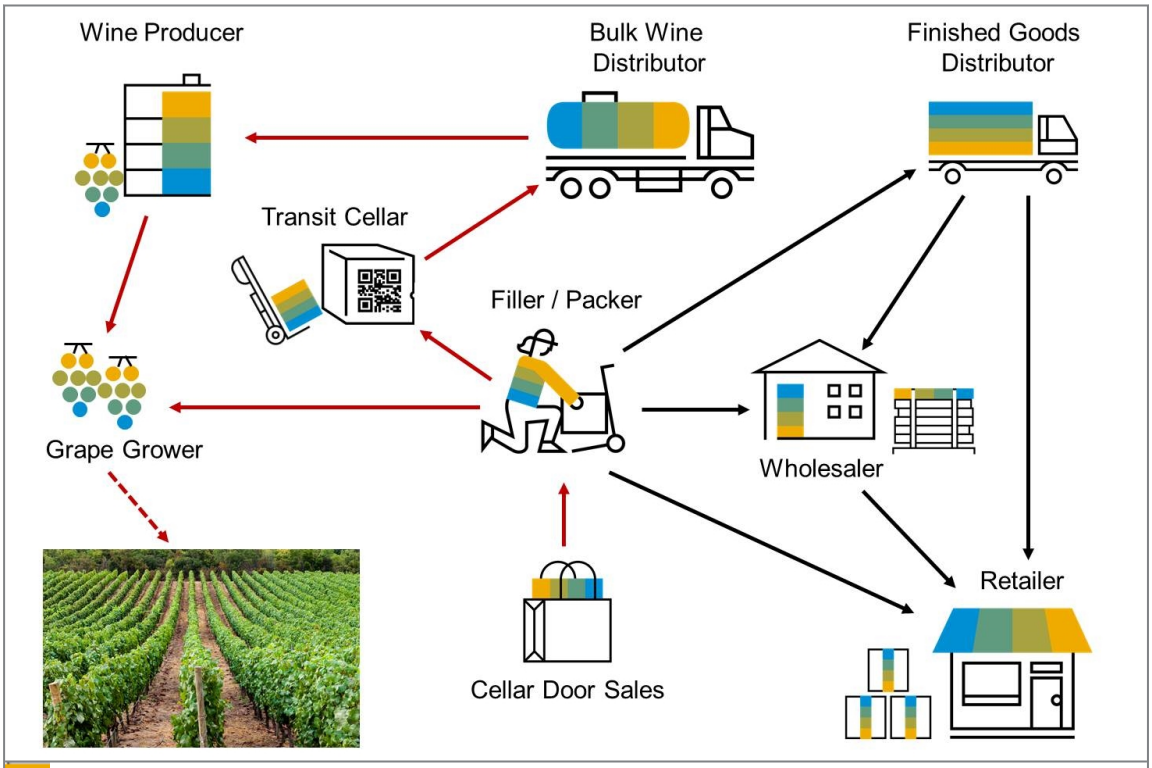


Figure 73: Graph Model: Example

Graph models can also indicate flow direction between entities and also any number of attributes can be added to nodes or the lines that connect them. This means that additional meaning can be added to the network and queries can be executed to ask questions relating to the network.

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 in the path
- What is the shortest path between point A and point B?

Graph processing allows you to discover hidden patterns and relationships in your network data, and all in realtime.

Graph Model: Example

There are many examples of where SAP HANA Graph could be used, 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.

- Text Analysis

  An SAP HANA Text Analysis stores the results in a flat table structure. But sometimes the individual words in text can have multiple relationships to the other words in the same text. Storing the results of a text analysis in a graph provides the optimal model for querying the relationship between words. Without a graph model you would have to create a separate row in the table to represent the relationship from one word to all other words.

It is possible to use standard SQL tables with standard data definitions and query code to create and process a similar model. However, it would be extremely complex to define such a model with SQL and also to query the model. Also, processing times could be challenging. SAP HANA Graph provides tools for graph definition and additional language for graph querying 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 using a dedicated in-memory graph engine.

Series Data

When you collect data at a measurable interval such as time, the data is called **series data**. Analysis of series data allows you to draw meaningful conclusions and predictions from the patterns and trends present in the values.

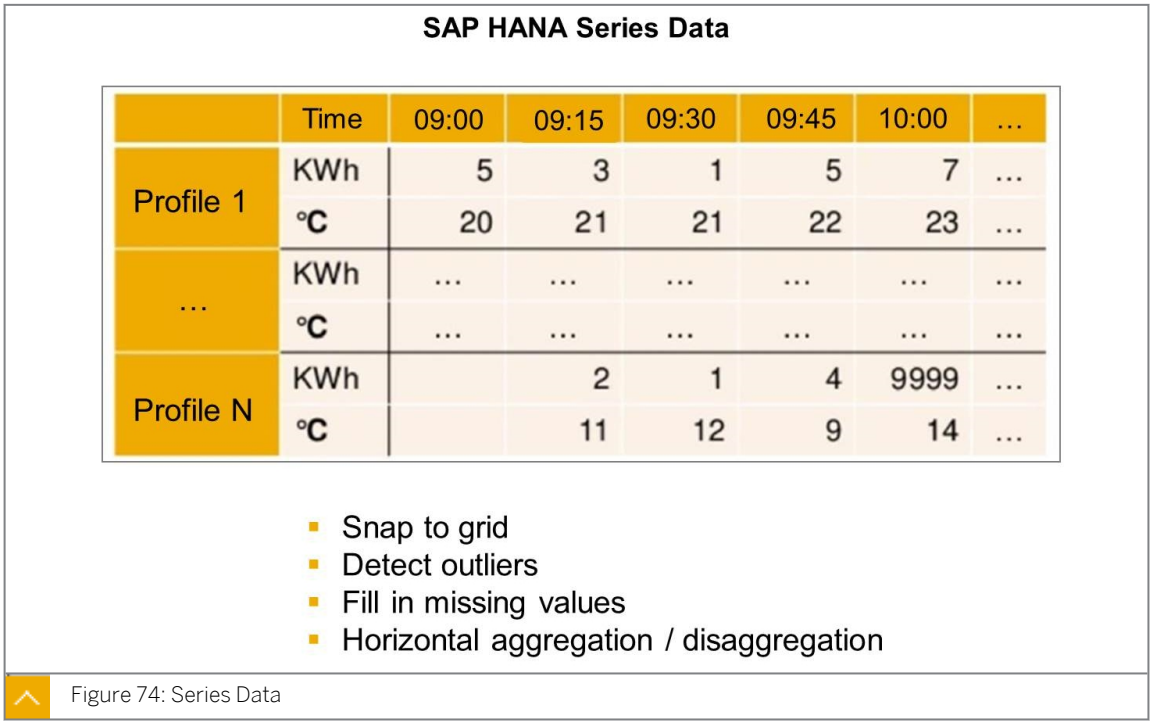


Figure 74: Series Data

A good example of series data that most people can relate to is when we consider energy metering. Energy companies are installing smart meters in households and businesses and are then able to collect energy consumption data at regular, and more frequent intervals that even before. SAP HANA Series Data provides efficient storage of this data using compression and also analytical capabilities on this data.

Some examples of the analytical capabilities of SAP HANA Series Data:

- Horizontal disaggregation — display meter readings that were collected at a course grain and disaggregate to a finer grain (for example, collect data at hourly readings and display are five minute intervals).
- Fill in missing values — if readings are missing, add them in.
- Detect abnormal readings.
- Adjust reading to nearest hour — when reading are not collected exactly on the hour, snap them to either the previous hour or the next hour depending on your rules.

There are many use cases for SAP HANA Series Data. Think of any scenario where data is collected at regular intervals, not necessarily relating to time intervals, maybe relating to an recurring event. The key word is **series**.

LESSON SUMMARY
You should now be able to:

- Advanced Analytics with SAP HANA

Unit 3
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

Connecting On-Premise tools to SAP HANA

A popular use case for SAP HANA is to power real-time business intelligence (BI).

SAP HANA can be deployed as a local data mart, to capture data from various operational systems, including SAP enterprise systems and non-SAP sources. Data can be loaded to SAP HANA either using regular batch loading or real-time updates. It is even possible to have SAP HANA connect to data sources in remote systems directly so that data loading to SAP HANA is not necessary. SAP HANA then become a logical layer sitting between the data sources and the reporting tools to provide the data model.

SAP HANA uses many industry standard connection protocols to connect to the database. SAP HANA can connect directly to a large number of reporting tools, including SAP Business Objects. So, reporting on SAP HANA can begin immediately.

For customers who run SAP enterprise suites such as Business Suite or SAP S/4HANA, one of the most appealing aspects of the SAP HANA BI 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 enterprise operational data, based on the tables of either SAP Business Suite or SAP S/4HANA. SAP has developed and maintained these comprehensive virtual data models to expose live operational tables from all areas of SAP Business Suite and SAP S/4HANA.

The real value in the virtual data models is the business semantics provided by SAP. Raw database tables are combined, filters, and calculations are added to expose business views ready for immediate consumption by any reporting tool with no additional modeling needed. So, instead of having to define the data models and semantics in your reporting tool, this is be done once in SAP HANA and shared by all tools.

As well as consuming the ready-made data models provided by SAP — either calculation views for Business Suite, or ABAP CDS for S/4HANA — you can create your own custom calculation views and consume these from any reporting tool.

Connecting Reporting Tools

Because SAP HANA provides database access using industry standard connectors, all SAP reporting tools are able to connect to SAP HANA directly using one or more of the following:

- ODBC — for generic relational connections

- JDBC — for JAVA-based relational connections
- ODBO — for multidimensional data source connections
- BICS — SAP proprietary connector only used by SAP reporting tools

Some reporting tools offer a choice of connectors. For example, with Crystal Reports you can use either a JDBC or ODBC connectors. Some connectors support more SAP HANA features.

The query language used depends on the reporting tool used and the connection type. The two database query languages used by SAP HANA are SQL (for relational models) and MDX (for multidimensional models).

SAP BI tools

Let's describe the current, recommended SAP BI tools:

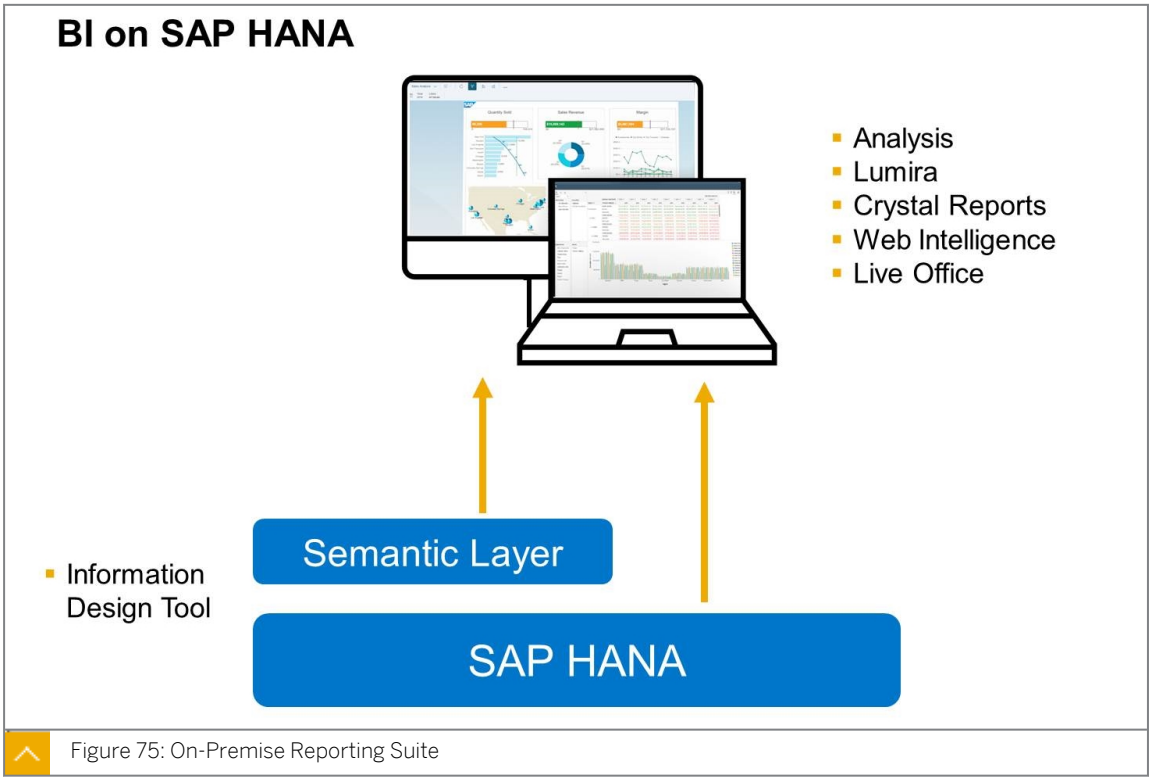


Figure 75: On-Premise Reporting Suite

- SAP Analysis, Edition for Microsoft Office — Powerful OLAP reporting in a well-known and popular Excel environment.
- Lumira — Build intelligent dashboard and cockpits that can be deployed automatically to mobile devices.
- Web Intelligence — End user, self-service query tool using a simple wysiwyg drag and drop interface.
- Crystal — Create consumer grade, professional looking enterprise reports that require precise layout and formatting.
- Live Office — Embed real-time Business Objects reporting content into Microsoft Office documents.

These tools can be implemented standalone or as part of an implementation of the powerful and popular Business Objects Platform, where user administration, security and report

storage and access can be centrally managed. With the Business Objects Platform, reports can also be scheduled. Using the bursting capabilities, report content can be broken up and distributed to various user groups.

Semantic Layer

All SAP reporting tools can connect to SAP HANA directly, however is it also possible to build an additional semantic layer into the BI stack.

Using the **Information Design Tool (IDT)** it is possible to create a sophisticated data model that combines multiple data sources (including SAP HANA) into a harmonized model. This is typically not needed where data integration is managed by SAP HANA. However, if data is integrated at a layer above SAP HANA, this approach might make sense.

> Note:
> The IDT tool is an evolution of the Business Objects Universe Designer. The Semantic Layer is the successor to the Universe.

SAP Analytics Cloud and SAP HANA

SAP Analytics Cloud (SAC)

Increasing numbers of organizations are moving their analytics to the cloud. SAP Analytics Cloud (SAC) provides a comprehensive tool set for developing and deploying BI content that is easily shared.

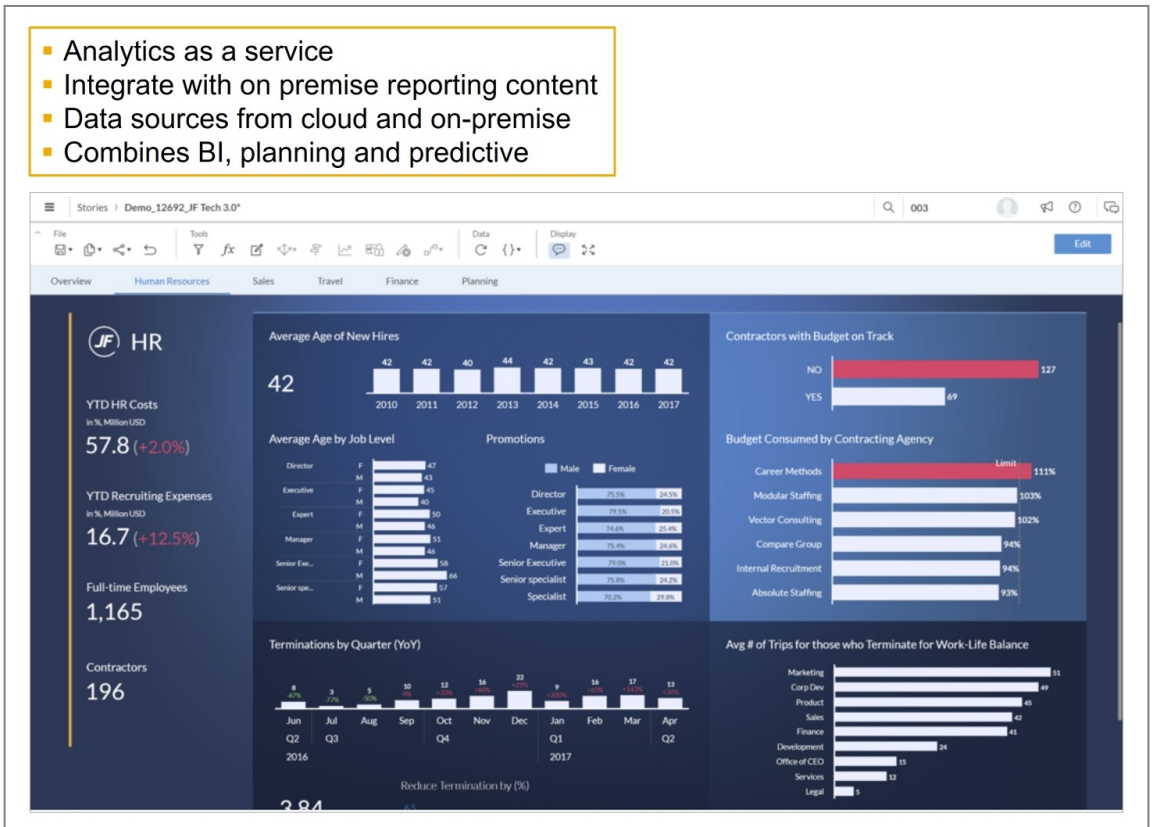


Figure 76: SAP Analytics Cloud

To support SAP's analytics strategy of **hybrid analytics** (on cloud mixed with on-premise analytics), SAC can take live data sources from on-premise (such as Universes, BW and SAP

HANA) and also cloud data. SAC also combines reporting content from on premise tools such as Web Intelligence and combine with cloud developments.

SAC provides a full suite combining BI, planning, and predictive capabilities and plenty of industry content is already available for quick start.

The SAP Analytics Cloud Digital Boardroom provides C level executives with a 360-degree views of all KPIs with the ability to drill down and explore the underlying data.

LESSON SUMMARY
You should now be able to:

- Connect SAP Business Intelligence Tools to SAP HANA

Learning Assessment

1. Why do we model in SAP HANA?

   Choose the correct answers.

   - ☐ A To push data-intensive processing away from the application and to the database.
   - ☐ B To develop reusable data processing logic in the database.
   - ☐ C To design an efficient data storage architecture.

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 write custom SQL to present a complex data source to a calculation view, which SAP HANA object do I use?

   Choose the correct answer.

   - ☐ A Procedure
   - ☐ B Table 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 Authorization Object
   - ☐ C Analytic Privilege

5. 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.

6. Why would an organization implement SAP HANA Spatial?

   Choose the correct answer.

   - ☐ A To enrich core ERP data with geographic insight
   - ☐ B To identify unhappy customers who need a response
   - ☐ C To determine how closely related two customers are

7. Which of these provide the algorithms that can be used with SAP HANA predictive analysis?

   Choose the correct answers.

   - ☐ A EML
   - ☐ B APL
   - ☐ C PAL
   - ☐ D SDA

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. Why would an organization implement SAP HANA Spatial?

   Choose the correct answer.

   - ☐ A To enrich core ERP data with geographic insight.
   - ☐ B To identify unhappy customers who need a response.
   - ☐ C To determine how closely related two customers are.

10. SAP HANA Graph Processing powers real-time, high-performance business charts, and dashboards.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

11. What are key capabilities of SAP HANA Analytics Cloud (SAC)?

    Choose the correct answers.

    - ☐ **A** Landscape monitoring
    - ☐ **B** Predictive
    - ☐ **C** Business Intelligence
    - ☐ **D** Planning

12. Which of the following does SAP HANA use to connect to reporting tools?

    Choose the correct answers.

    - ☐ **A** IDOC
    - ☐ **B** ODBC
    - ☐ **C** JDBC
    - ☐ **D** ODBO

Learning Assessment - Answers

1. Why do we model in SAP HANA?

   Choose the correct answers.

   [X] A To push data-intensive processing away from the application and to the database.

   [X] B To develop reusable data processing logic in the database.

   [ ] C To design an efficient data storage architecture.

   Correct — We push data-intensive processing away from the application and down to the database to obtain the best performance. Core modeling in HANA encourages reuse of data processing logic. Core modeling is used to develop calculation views for to create a virtual layer. Data storage in not included in the virtual layer.

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

   Correct — Dimension with star schema is the only type that is not valid.

3. If I need to write custom SQL to present a complex data source to a calculation view, which SAP HANA object do I use?

   Choose the correct answer.

   [ ] A Procedure

   [X] B Table Function

   Correct — Procedures cannot be used to present a data source to a calculation view, only table functions can do that.

4. What do we implement to restrict access to specific data rows of a calculation view?

   Choose the correct answer.

   - [ ] A SQL Permission
   - [ ] B Authorization Object
   - [x] C Analytic Privilege

   Correct — SQL permissions only secure the database object, not data values. Authorization object is an ABAP object, not used in SAP HANA. Analytic Privilege is used to define security around rows (data).

5. 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.

   Correct — *Voice of Customer* is a standard dictionary used in sentiment analysis that can identify expressions that customers may use to describe how they feel about products or services.

6. Why would an organization implement SAP HANA Spatial?

   Choose the correct answer.

   - [x] A To enrich core ERP data with geographic insight
   - [ ] B To identify unhappy customers who need a response
   - [ ] C To determine how closely related two customers are

   Correct — SAP HANA Spatial enriches ERP data with geographic insight. Unhappy customers would be identified using Text Analysis. How closely related customers are, is something you could find using SAP HANA Graph.

7. Which of these provide the algorithms that can be used with SAP HANA predictive analysis?

   Choose the correct answers.

   - [X] A EML
   - [X] B APL
   - [X] C PAL
   - [ ] D SDA

   Correct — EML, APL and PAL can all provide algorithms for SAP HANA predictive analysis. Another approach, not mentioned, is to use the R language for custom algorithm building. SDA is used for defining virtual tables that connect to remote sources.

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.

   Correct — *Voice of Customer* is a standard dictionary used in sentiment analysis that can identify expressions that customers may use to describe how they feel about products or services.

9. Why would an organization implement SAP HANA Spatial?

   Choose the correct answer.

   - [X] A To enrich core ERP data with geographic insight.
   - [ ] B To identify unhappy customers who need a response.
   - [ ] C To determine how closely related two customers are.

   Correct — SAP HANA Spatial enriches ERP data with geographic insight. Unhappy customers would be identified using Text Analysis. How closely related customers are, is something you could find using SAP HANA Graph.

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

    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.K

11. What are key capabilities of SAP HANA Analytics Cloud (SAC)?

    Choose the correct answers.

    - [ ] **A** Landscape monitoring
    - [x] **B** Predictive
    - [x] **C** Business Intelligence
    - [x] **D** Planning

    Correct — Landscape monitoring is not a capability of SAC. Predictive, Business Intelligence, and Planning are capabilities of SAC.

12. Which of the following does SAP HANA use to connect to reporting tools?

    Choose the correct answers.

    - [ ] **A** IDOC
    - [x] **B** ODBC
    - [x] **C** JDBC
    - [x] **D** ODBO

    Correct — IDOC is an SAP NetWeaver technology used to pass data between SAP systems.

UNIT 4 Data Management with SAP HANA

UNIT OBJECTIVES

- Describe Data Tiering with SAP HANA
- Describe SAP HANA data acquisition solutions

Unit 4
Lesson 1

Data Tiering with SAP HANA

LESSON OVERVIEW
In this lesson we will explain what is meant by Data Tiering.

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

- Describe Data Tiering with SAP HANA

Data Tiering

Data Temperatures

Today, it is possible to size hardware so that even the largest enterprise databases could be stored completely in memory. But it does not make sense to store old, infrequently accessed data in expensive memory. Memory should not be used as an archive and should be reserved for active data where instant access is needed. There are better solutions for managing data that is infrequently accessed at a lower cost than memory, but still providing good read performance.

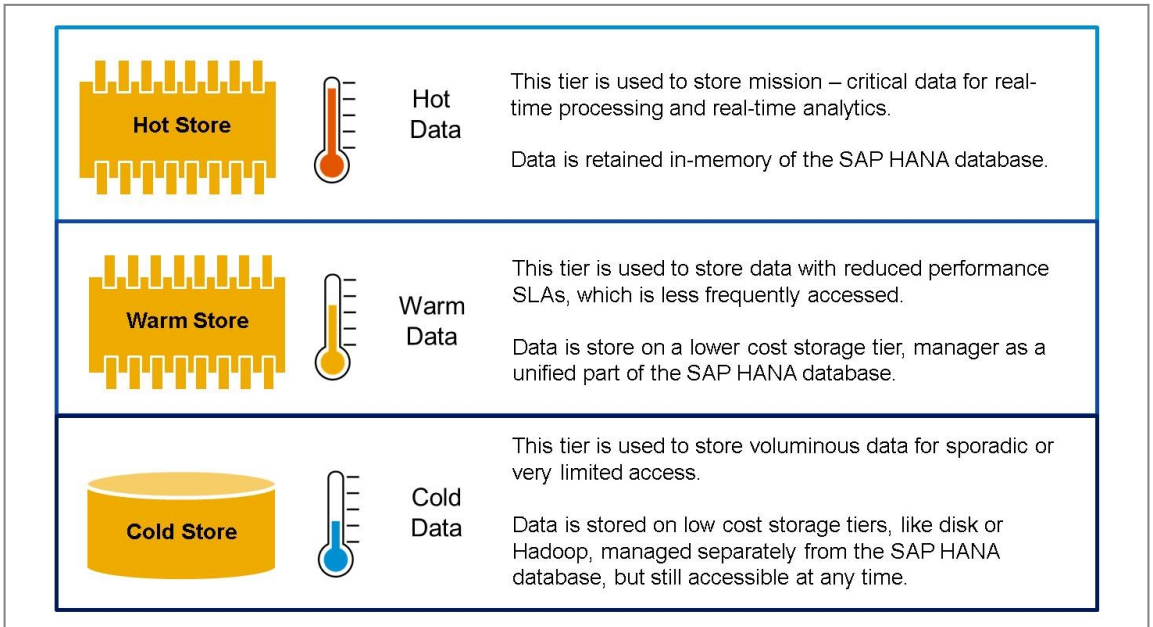


Figure 77: Data Tiering Layers

Data has a lifecycle. Typically new data is used often and as time passes, that data is accessed less. So it is helpful to classify data into a hierarchy of temperatures that relate to the speed of access required. SAP HANA provides a solution to handle three temperatures of data: hot, warm and cold. It is not just data aging that determines how it should be classified.

Some data does not requires fast access even though it might be new data. So warm or cold temperatures could also apply even to new data.

You should size HANA memory according to your hot data usage. This is an important calculation that can ensure that you get the right performance at the right price. Then, you implement cheaper tiers such as disk and archives for the older data or data that does not require fast access. But an important point is that all tiers in SAP HANA provide data that is always available to applications, just at different levels of performance. You also need to think about Service Level Agreements (SLA) that determine the expected performance of applications. Data read-access plays a major part in the performance of an application and is often the bottleneck.

You must also think about the type of application that you will run on SAP HANA, because there are many technical solutions offered by SAP to handle data storage across the temperature hierarchy, but not all application types are compatible with all tiering solutions.

The three application types are:

- Native HANA applications
- SAP BW and SAP BW/4HANA
- SAP Suite on HANA and SAP S/4HANA

,

Hot Tier

Data classified as hot is accessed frequently and/or needs very high performance. The storage basis for hot data is the HANA memory.

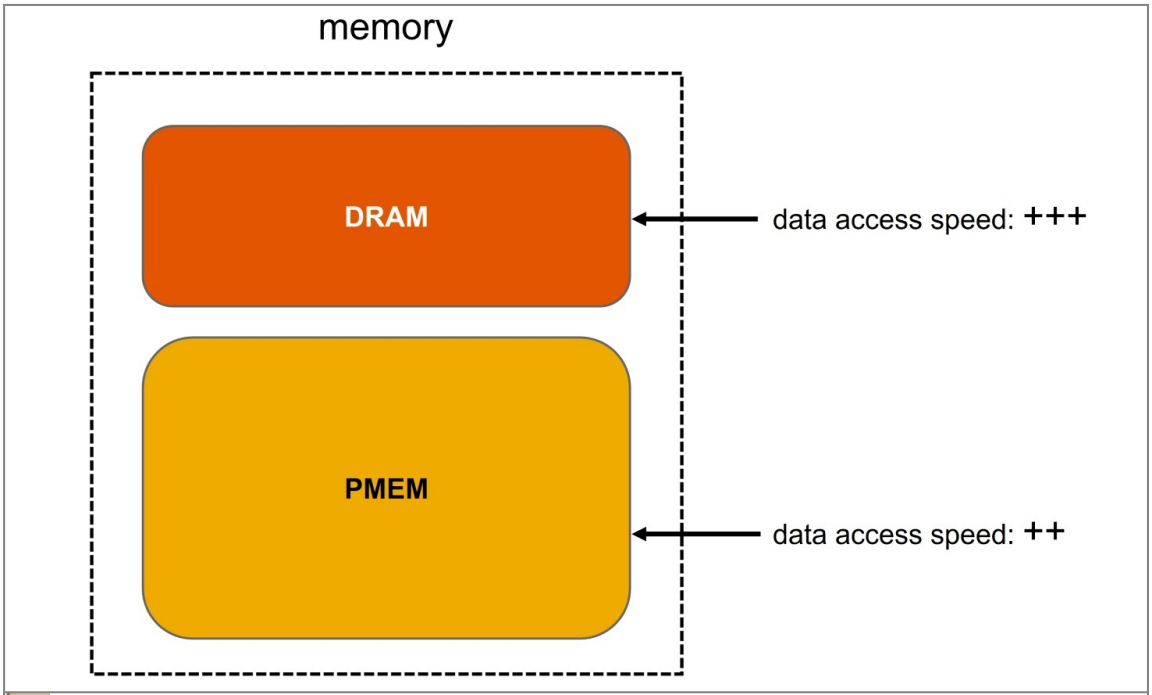


Figure 78: Hot Data Tier

But with the introduction of persistent memory (PMEM) alongside DRAM, there is actually a dual tier even within memory. Though not an official term, you could imagine the very fast DRAM might be referred to as red-hot data and the slightly slower PMEM as simply hot. You decide on the ratio between DRAM and PMEM by following SAP guidelines. PMEM is treated

just like DRAM and the distribution of data between DRAM and PMEM is handled automatically by SAP HANA.

Note:
DRAM and PMEM are compatible with all three types of application.

Warm Tier

There are three solutions in SAP HANA that support warm data. The most recently introduced solution is called **Native Storage Extension** (NSE).

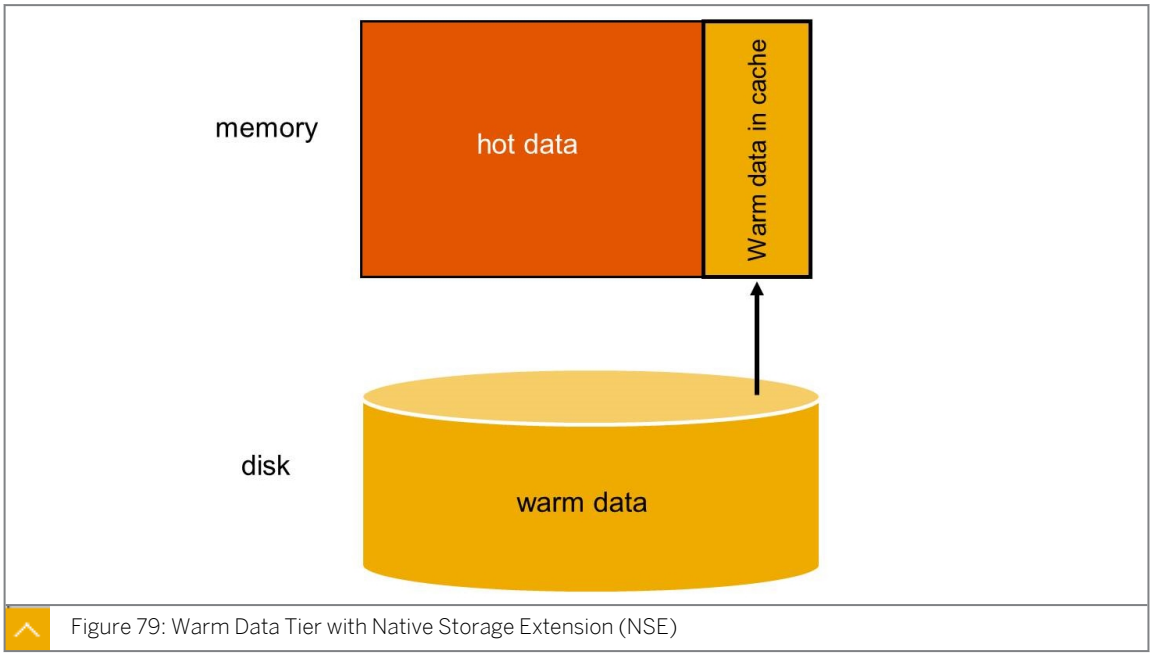


Figure 79: Warm Data Tier with Native Storage Extension (NSE)

NSE primarily stores data on a persistency layer (this could be disk or SSD) and uses an intelligent buffer cache to load and unload data to and from memory based on usage patterns. The persistence layer used by NSE is the same persistence layer used by memory when unloads are required due to memory becoming full.

Initially, you allocate data to NSE at either the table level, partition level or a column level. Data is loaded to the buffer cache as needed by the chosen unit of allocation. For example, if you have chosen to allocate an entire table to NSE, when you need any data from that table then the entire table is loaded to buffer cache from the persistent layer.

The buffer cache in memory will intelligently unload pages of data back to the persistence layer when it is full.

One of the key benefits of NSE is that it is fully integrated into the HANA database. For example, the persistent storage is the same that is used by the memory, so all operations such as back ups are automatically including in this warm tier solution.

The other two warm storage solutions are **Extension Node** and **Dynamic Tiering** and these have been available now for some time and are embedded in many customers' SAP HANA landscapes.

Extension Node — In any SAP HANA landscape there is always a worker node that is the primary node dealing with data processing. It is possible to add extension nodes that can support the worker node to provide additional processing capacity using parallelization

techniques. But you can also add an extension node to use for warm data storage. One of the key reasons for choosing this solution is that the extension node solution is currently the fastest solution for warm data access as it stores data in column based tables in memory and not on disk. The downside is that it is also the most expensive to implement as it requires additional hardware resources to host the warm data extension node.

Note:
Extension node for warm storage solution is supported by native HANA applications and also BW / BW4HANA but is not supported for Suite on HANA or S/4HANA.

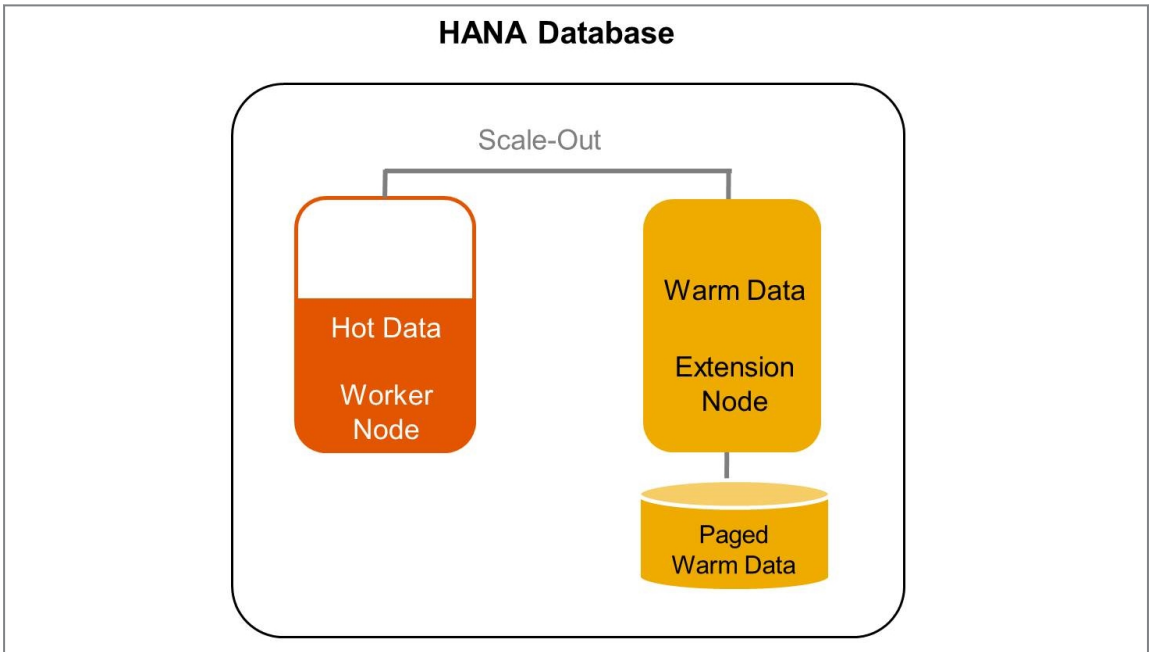


Figure 80: Warm Data Tier with Extension Node

Dynamic Tiering — Dynamic Tiering (also known as *Extended Storage*) can be regarded as the forerunner to NSE. It is still fully supported and its appeal is that it is currently the largest warm data storage solution and can handle up to 100 TB of data in Dynamic Tiering.

Whenever you create tables, you specify whether the table should have its primary residency in memory (hot) or in the persistent layer (warm). The persistence storage is a separate installed component that sits along side the core in-memory database. You can split tables across the hot and warm tiers by partitions. So for example, you could assign the current year to hot storage for fast access and other years to warm storage. The name given to a table that has partitions across warm and hot storage layers using Dynamic Tiering is a **multi-store table**.

With NSE you can also split tables by partitions across hot and warm tiers, but the key difference is that Dynamic Tiering uses its own persistent storage layer for the warm data whereas NSE uses the same persistent layer as the in-memory database. So the term multi-store table only applies to Dynamic Tiering because we are splitting data across two separate *stores* of data. The term multi-store table does not apply to NSE, though the concept does apply.

Note:
Dynamic tiering is supported by native HANA applications but not by BW / BW4HANA and not by Suite on HANA or S/4HANA.

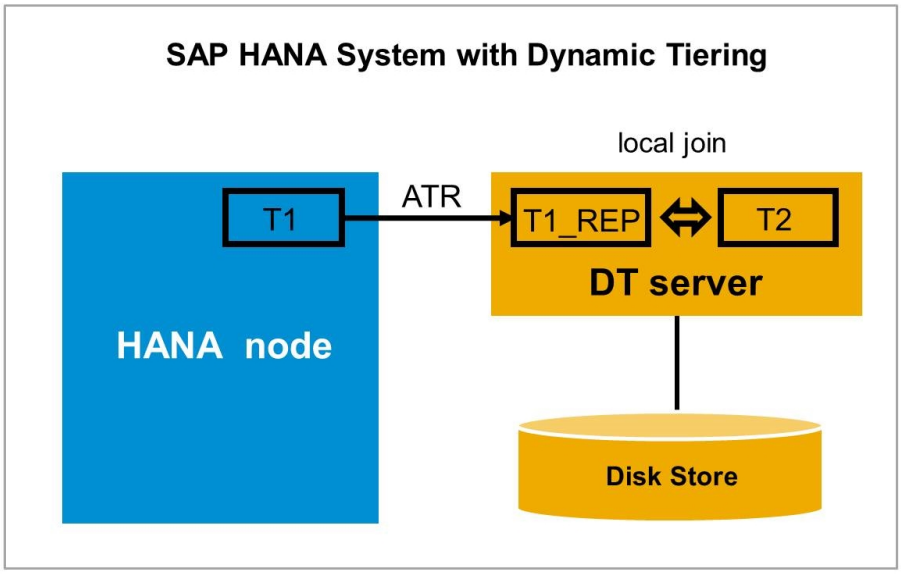


Figure 81: Warm Data Tier with Dynamic Tiering

So when we consider the three warm data storage solutions we need to think about the compatibility of our application and then the **cost** of implementing and running each solution, the **performance** it provides and also the ability to cope with growing **data volumes**.

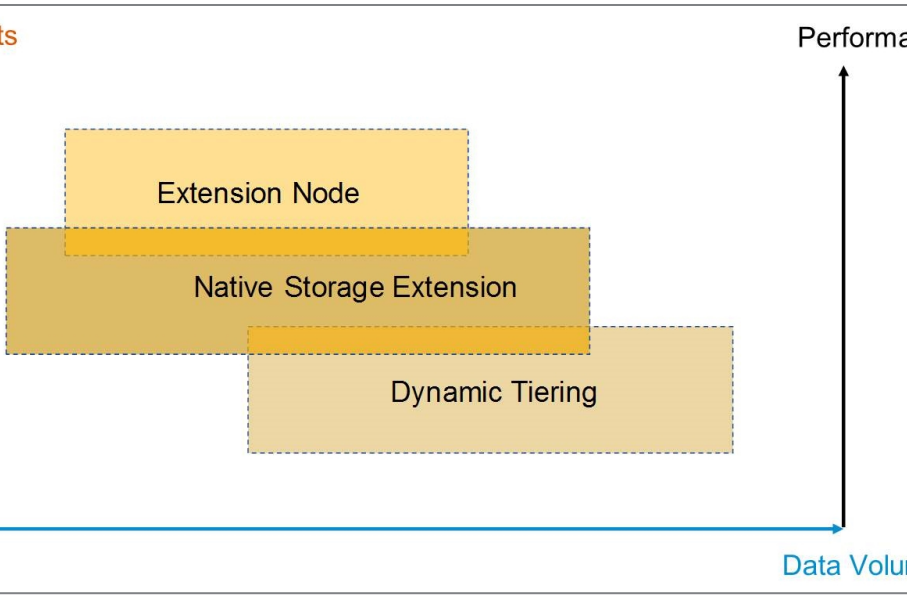


Figure 82: Comparing Warm Tier Solutions

SAP recommend that customers should always evaluate NSE before looking at the other warm storage solutions. It is not recommended to mix warm data storage solutions.

Cold Tier

There are two solutions for cold data storage.

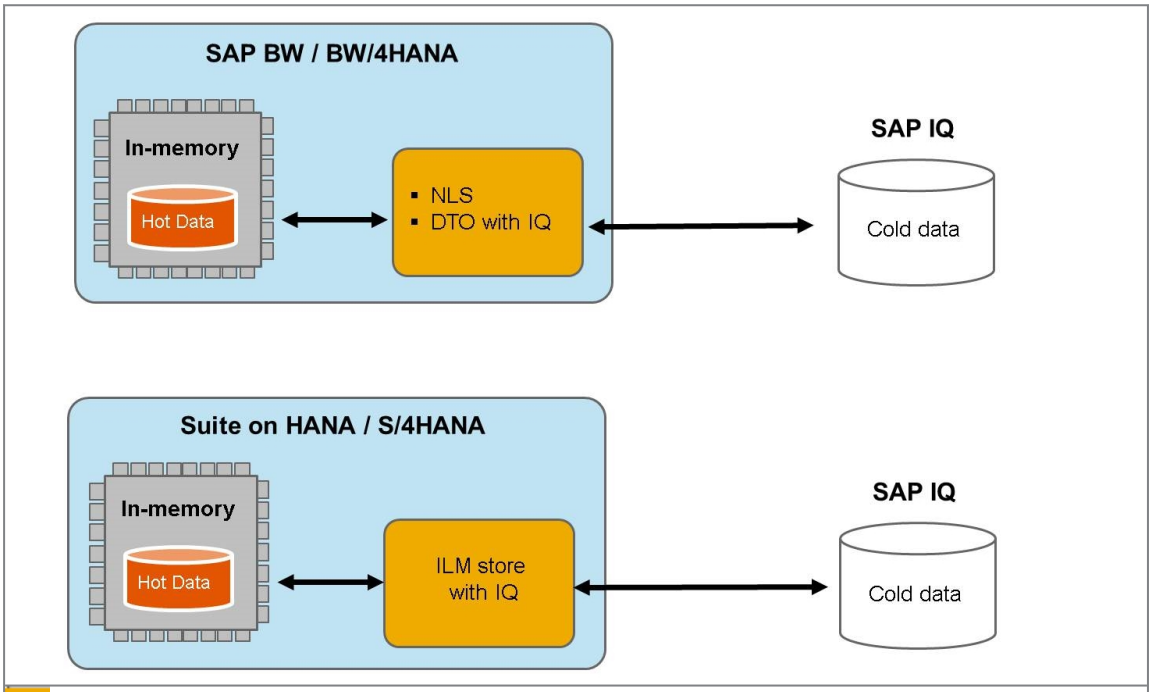


Figure 83: Cold Data Tier with SAP IQ

SAP IQ — This cold data storage solution utilizes an external SAP IQ column store database connected to SAP HANA to provide very large data storage of data that is infrequently used. SAP IQ is used by BW on HANA and BW/4HANA via the NLS interface. SAP IQ is used by Suite on HANA and S/4HANA to store the ADK files only. SAP IQ is not used by native SAP HANA applications.

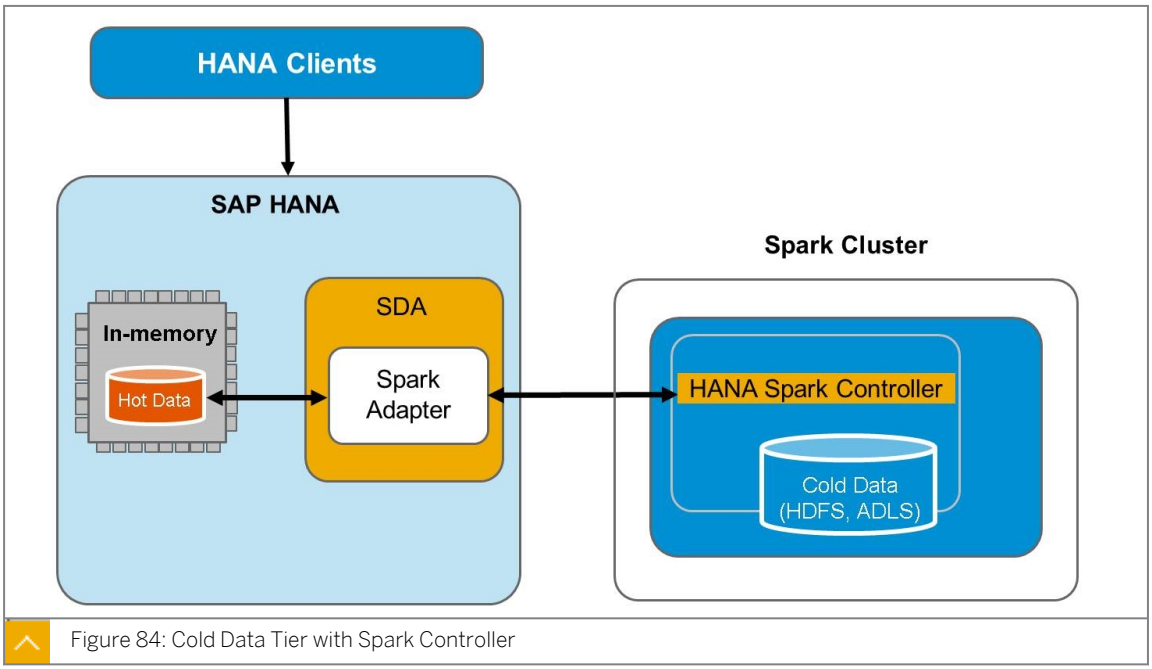


Figure 84: Cold Data Tier with Spark Controller

Spark Controller — The Spark Controller sits in the chosen cold data storage system (for example, HDFS / ADLS / Cloud) and communicates with SAP HANA via SDA to reach data. For native HANA applications and also BW and BW/4HANA you can use the Spark Controller solution with your chosen data storage vendor. For Suite on HANA and S/4HANA we can only store ADK files in HDFS.

LESSON SUMMARY

You should now be able to:

- Describe Data Tiering with SAP HANA

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 Acquisition Concepts

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; in real time or batch.

SAP HANA is able to extract and store the captured data but it can also create connections to remote data sources so that remote data source can be exposed to SAP HANA for live data access.

You need to decide which data acquisition tools are required. There are many to choose from and each has its strengths. For example, some tools have their strengths in complex ETL, whereas other tools can handle optimal fast real-time replication. Some tools can even handle a combination of both. For data that is loaded to SAP HANA, you also need to decide where the data resides. There are multiple options for specific data locality in SAP HANA that are suited to the different data temperatures.

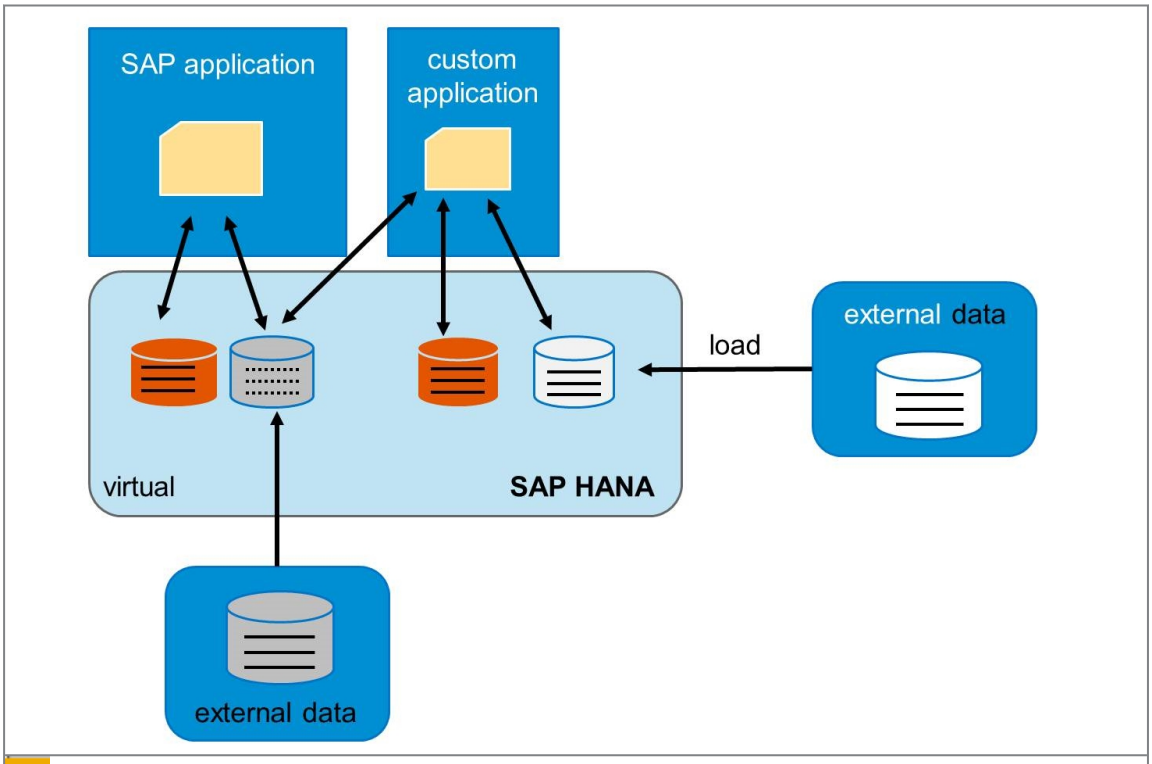


Figure 85: Data Acquisition

Before we dive into the data acquisition tools, let's look at a scenario where data acquisition tools may not even be needed.

Applications that are powered by SAP HANA generate SQL statements that create, update, and delete records in the SAP HANA database. In this case, no additional data acquisition tools are needed. This is true for SAP applications such as SAP S/4HANA and BW and also for custom applications that run on SAP 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 SAP S/4HANA or BW, you may want these applications to have access to additional data sources, for example adding textual data from Twitter to a sales opportunity created in SAP 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. Another term that is being more frequently used in place of data acquisition is data ingestion.

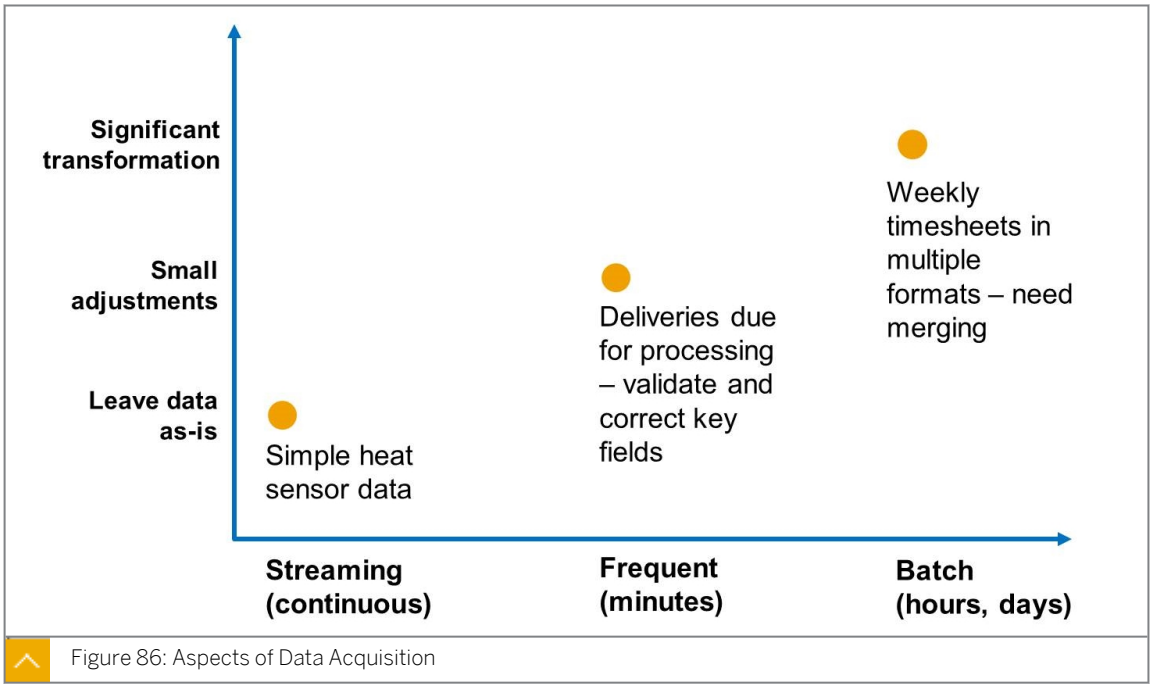


Figure 86: Aspects of Data Acquisition

When you consider data acquisition, think about the following key aspects:

- Frequency of data acquisition
- Transformation and cleansing requirements of data
- Integration requirements when there are multiple data sources that need to be combined

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 events, such as when a vending machine runs out of stock and transmits a request for a refill.

Transformation and integration is represented on the vertical axis of the figure, *Aspects 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 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 adapters 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

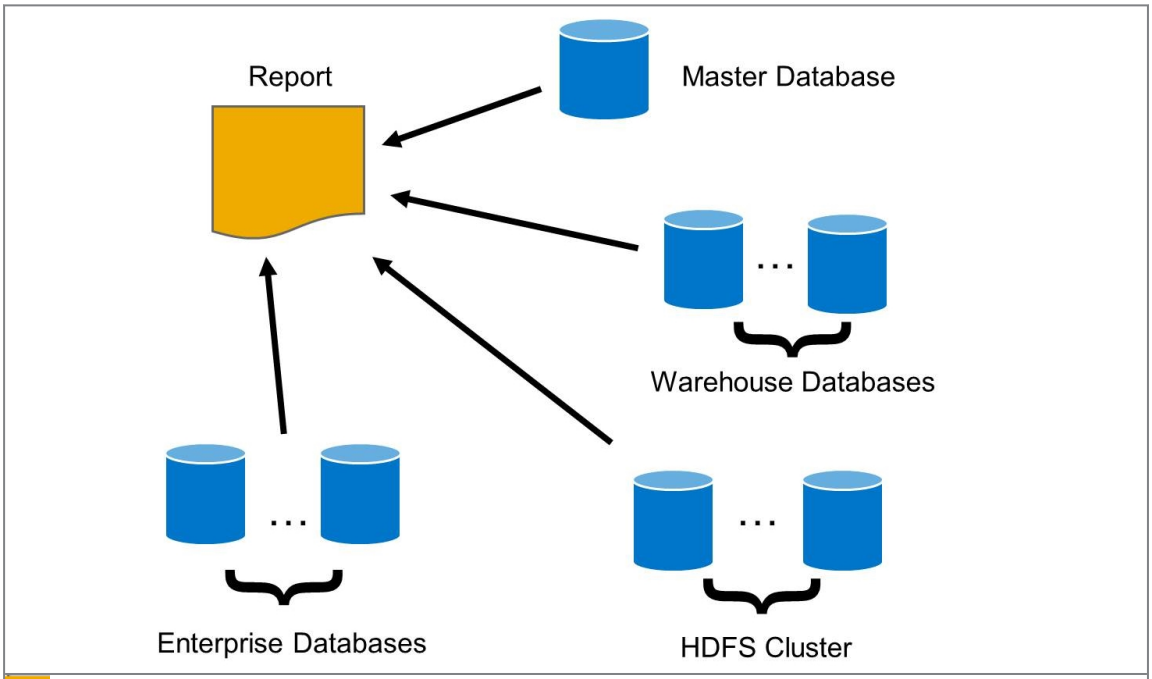


Figure 87: Motivation for Smart Data Access

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.

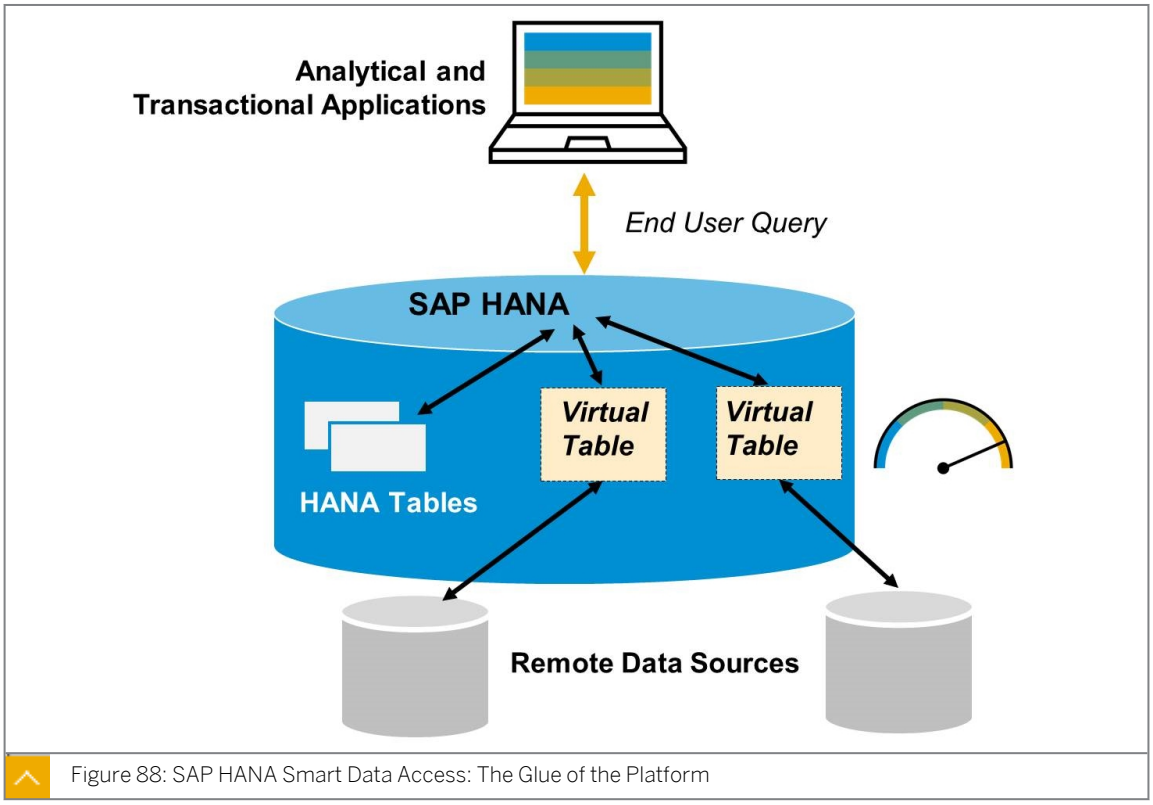


Figure 88: SAP HANA Smart Data Access: The Glue of the Platform

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 the metadata changes in the remote database, it can be easily re-synced with the HANA virtual table. This re-syncing 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 examples 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.

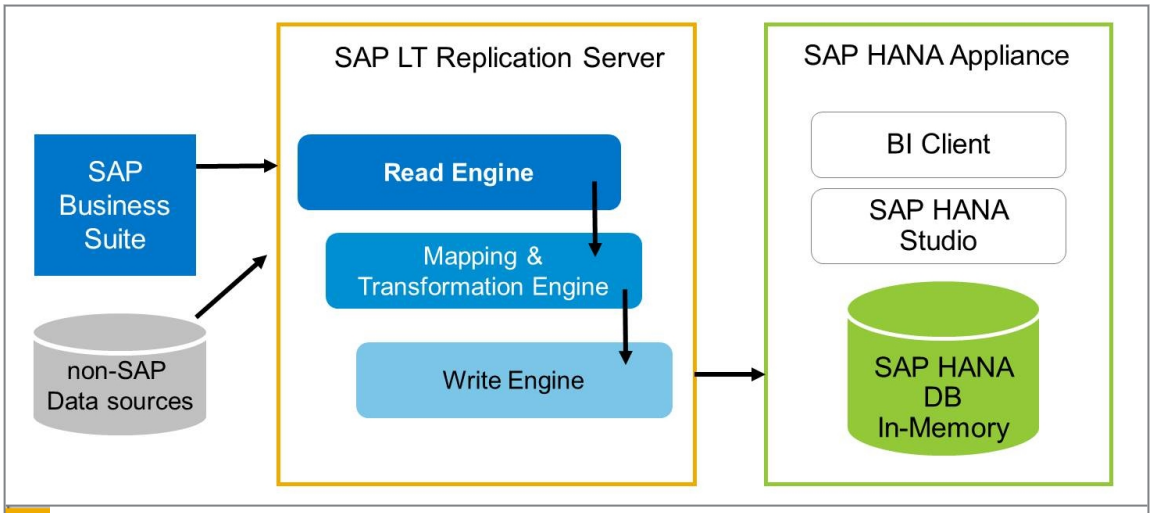


Figure 89: SAP LT Replication Server

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 moves 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 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 (for example, 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 are made.

Any transformation applied to data as it is being replicated has an effect on the time it takes for the data to arrive at the target. For this reason, only light transformations are 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 SAP 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.

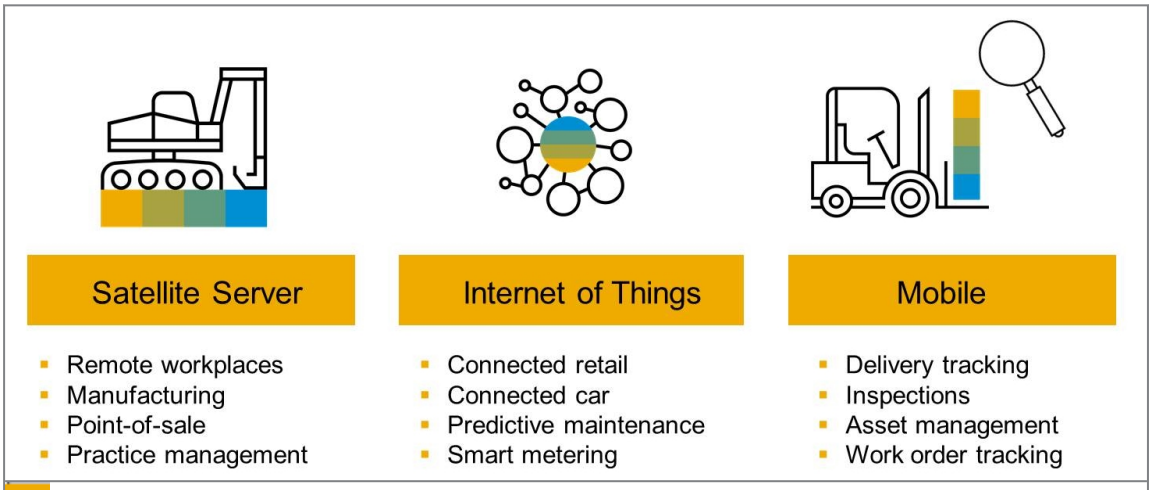


Figure 90: SAP HANA Remote Data Sync

SAP HANA remote data sync is useful when applications cannot remain continually connected to a central database due to connection problems. An 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.

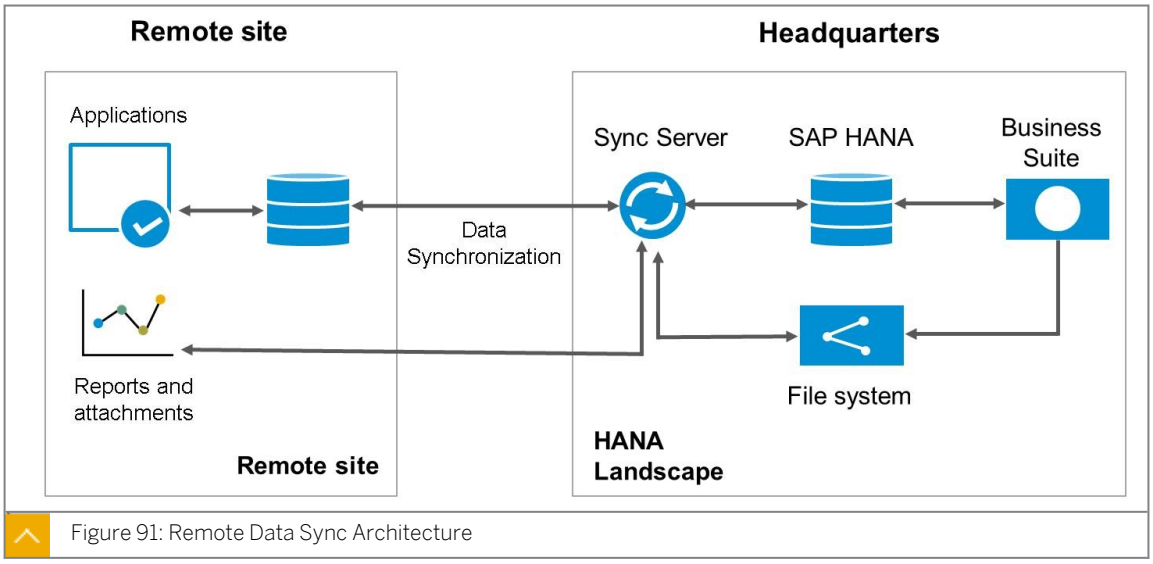


Figure 91: Remote Data Sync Architecture

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 set up 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.

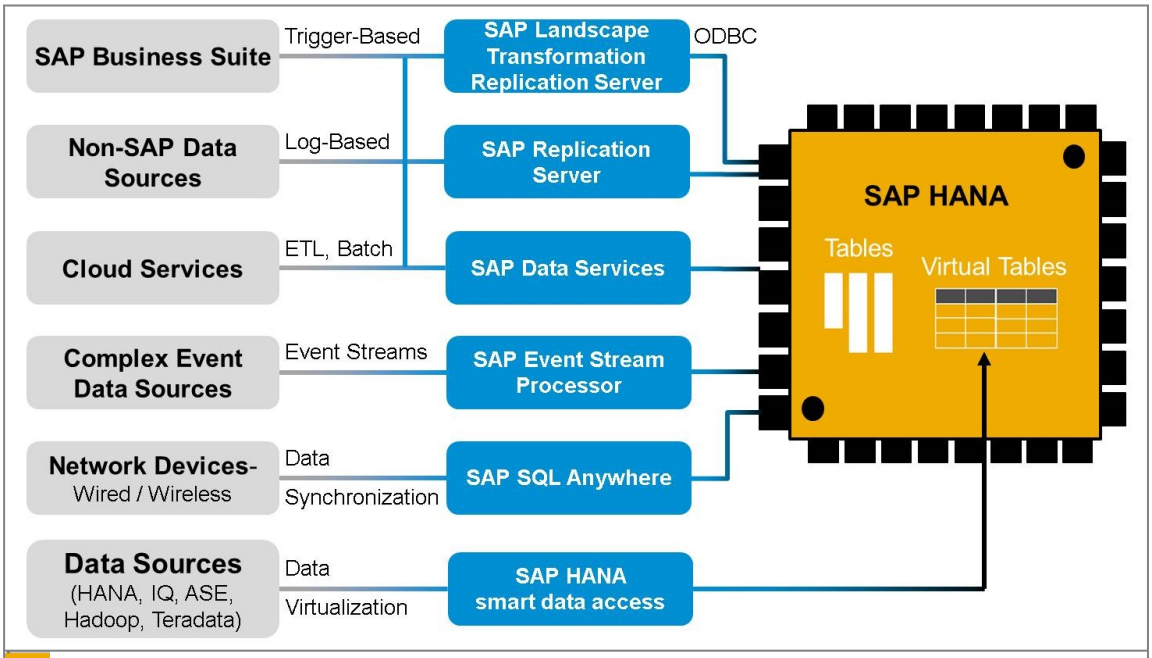


Figure 92: Traditional Approach to EIM

The inclusion of an additional data provisioning tier adds to the complexity of the overall solution.

A Simplified Approach to Data Integration

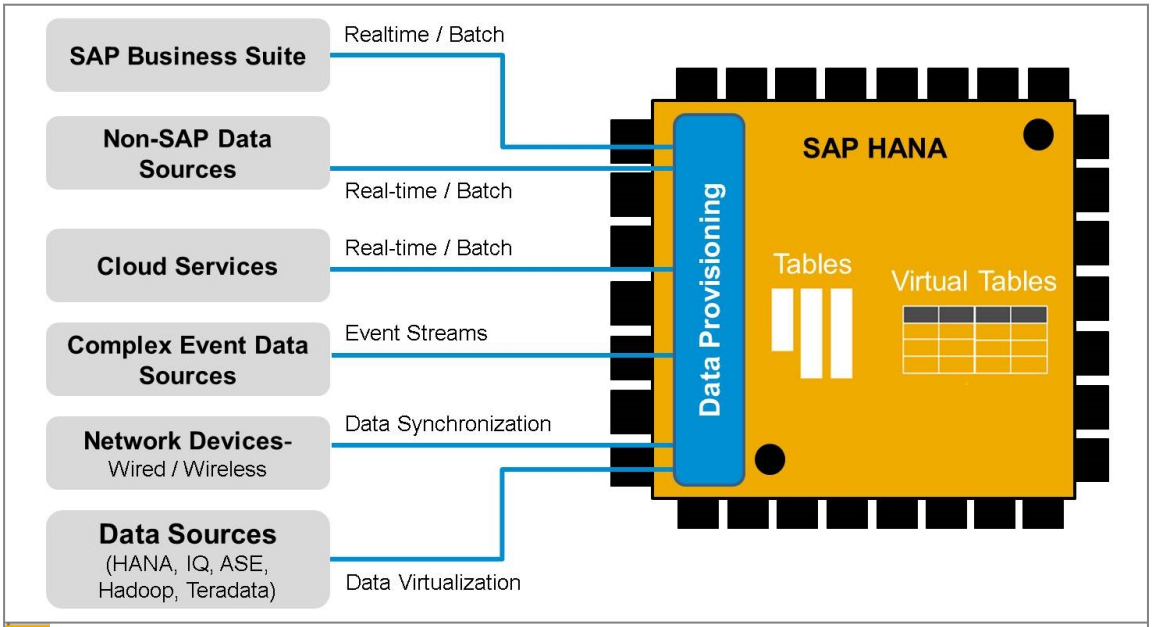


Figure 93: Simplified Approach to EIM

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

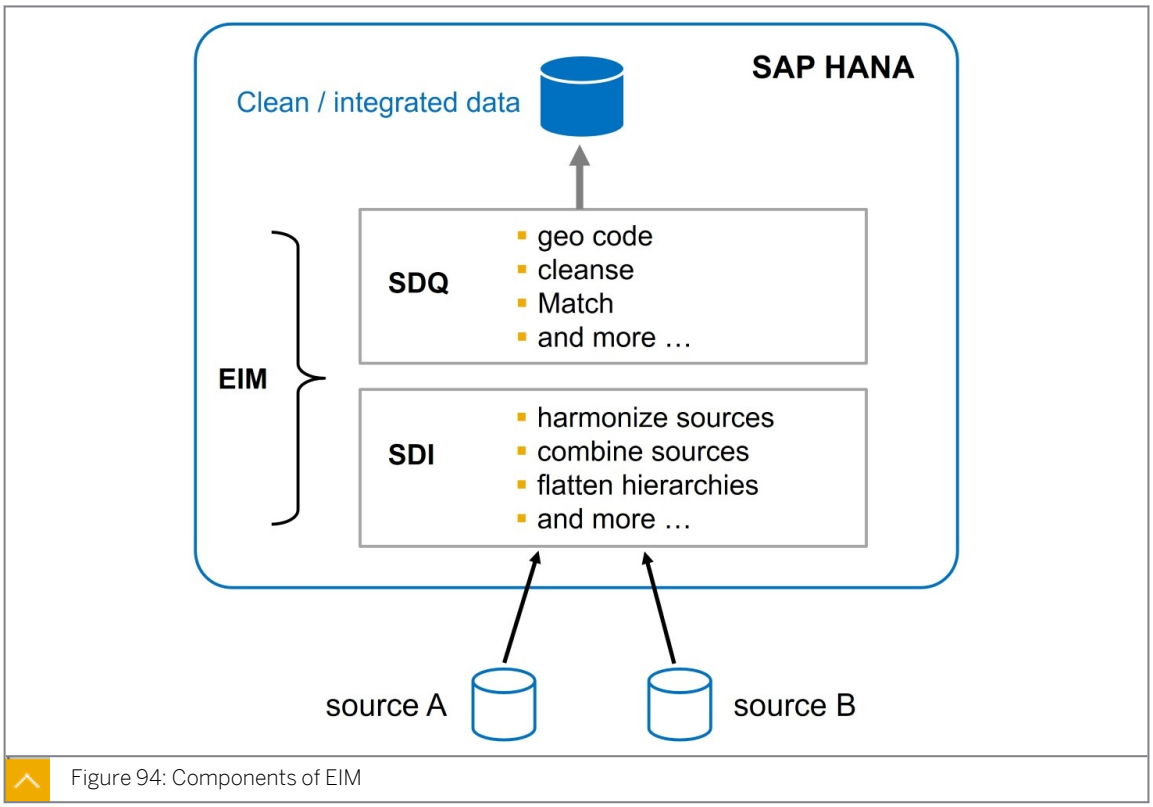


Figure 94: Components of EIM

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 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.

- Transformations enriching data
- Cleanse
  Parse, standardize and enrich person, title, phone, firm, email and address information within a specified input source.
- Geocode
  Enrich address data with associated latitude and longitude information

---

- **Simplify cleanse transforms:**
  Single transform deals with
  - Person names and titles, phone, email,
  - Firm,
  - Address information

  In Data Services, it is in 2 transforms
- **Consolidate available configuration options:**
  Improved productivity at functional parity

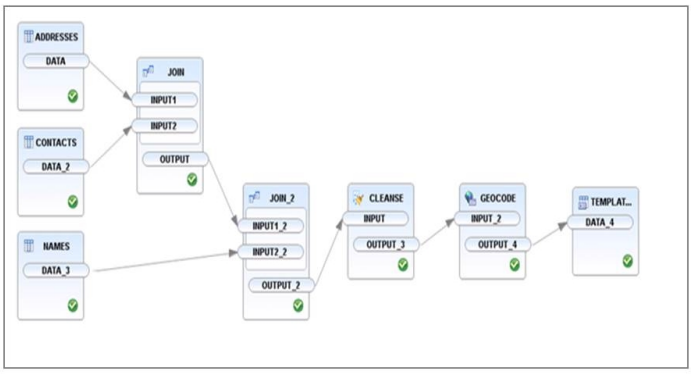


Figure 95: Building an EIM Data Flow

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.

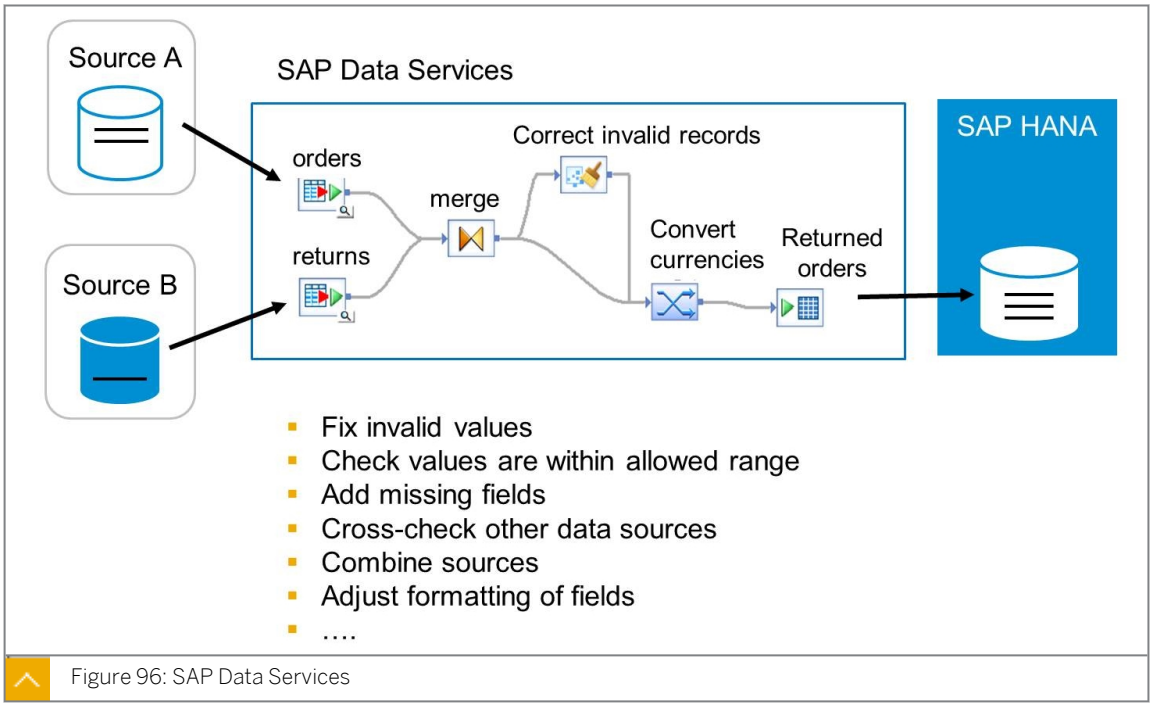


Figure 96: SAP Data Services

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 has 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.

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.

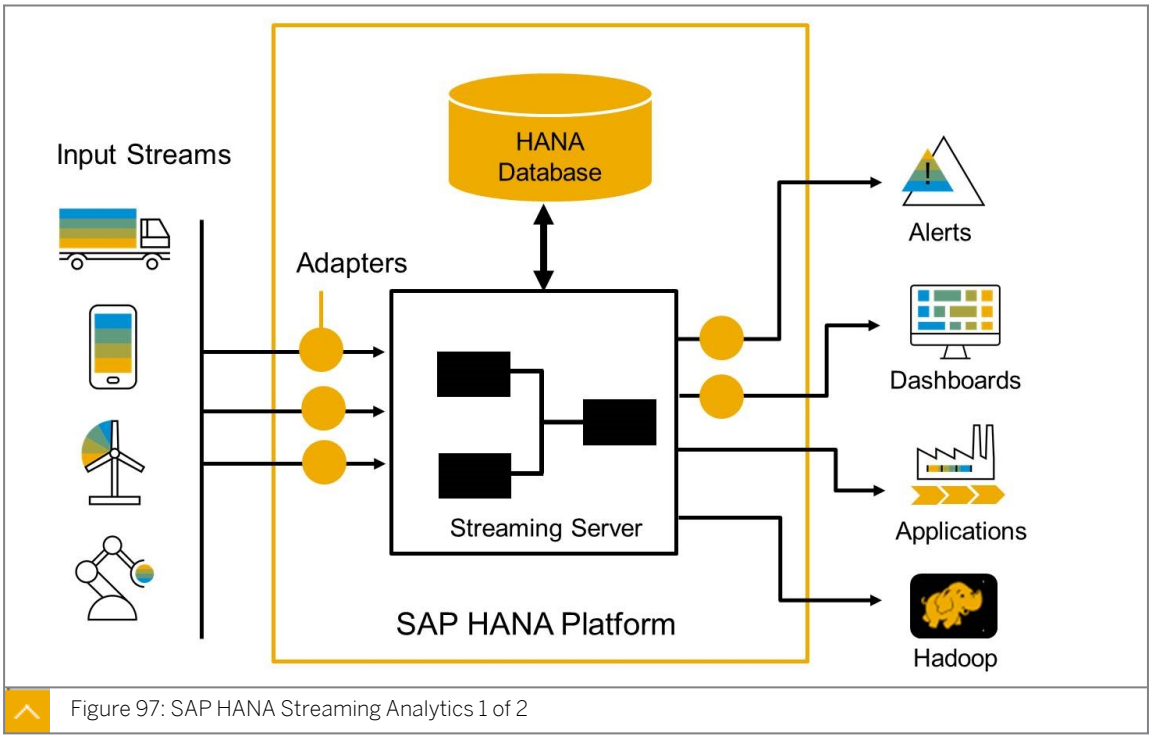


Figure 97: SAP HANA Streaming Analytics 1 of 2

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.

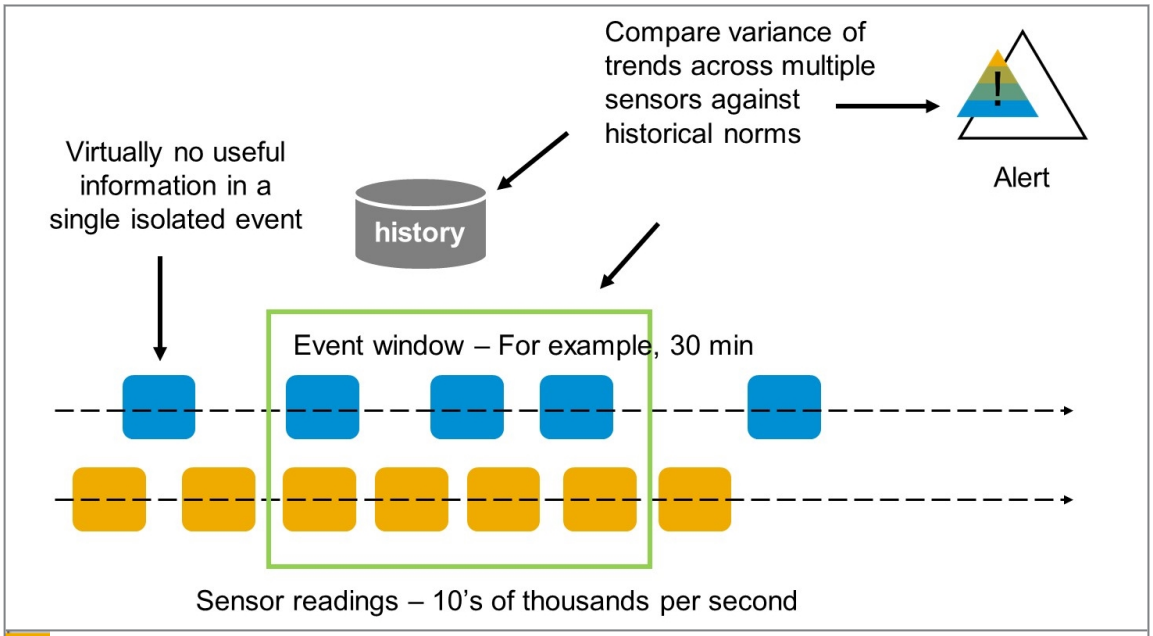


Figure 98: SAP HANA Streaming Analytics 2 of 2

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

Flat files can be used to load column or row tables in SAP HANA. There are two main approaches that you can choose depending on how repeatable the file load should be.

1. Using Design-time Files to Specify the Flat File Load

1. Create the target table (if it does not already exist) using the core data services (CDS) source file type *.hdbcds* to specify the table meta data.
2. Add the flat file data to your project database module using the source file type *CSV*.
3. Define the load parameters using the source file type *.hdbtabledata* which specifies the source file and target table, which columns are required, and the loading parameters, such as the delimiter symbol, whether a header exists, and so on.
4. In the *Development* view of Web IDE, build all the files, which also executes the load on a successfully build.
5. Check the target table now contains the loaded records.

The key benefit to using the design-time files to define all aspect of the table load is that you are easily able to redeploy the load in any target systems as you have described all the components in re-usable source files.

2. Web IDE Import Data Wizard

The Database Explorer of the Web IDE provides a data import wizard that is easy to use and is aimed at one-time data loads from flat files. This tool is ideal for when you quickly need some data for testing or development tasks.

CSV, XLS and XLSX files are supported. You can either load to an existing column or row table or have SAP HANA automatically create the table (row or column) during the data import based on the source file fields.

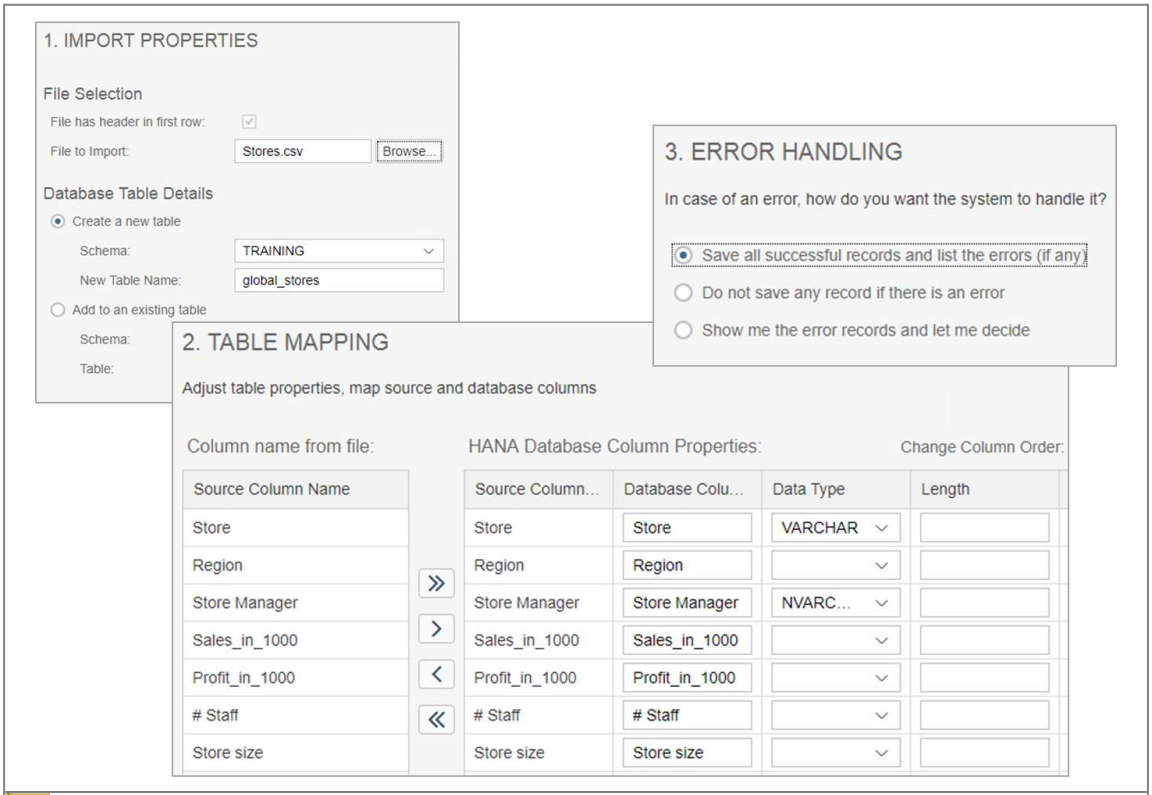


Figure 99: Import Flat File using Web IDE Import Data Wizard

The steps to follow are:

1. Select the source file (csv, xls,xlsx).
2. Select an existing SAP HANA table or choose the name of the new table that you would like to create.
3. Map the fields of the source file to the columns of the target table:
   - For new target tables you can choose the column type and length and also the table type (row or column).
   - For new or existing tables you can choose which source file fields you would like to load, you can ignore fields you do not need.
4. Choose the error handling settings.
5. Execute and check the target table contains the loaded records.

Note:
Creating new target tables during the import is possible only for classic catalog schema-based tables and not HDI container-based tables.

LESSON SUMMARY
You should now be able to:

- Describe SAP HANA data acquisition solutions

Learning Assessment

1. In SAP HANA, cold data is stored in HANA Native Storage Extension (NSE).

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. What are typical characteristics of SAP HANA data replication?

   Choose the correct answers.

   - ☐ **A** Virtualization
   - ☐ **B** Duplicate data
   - ☐ **C** Real-time
   - ☐ **D** Continual streaming
   - ☐ **E** Significant transformations

3. SAP SLT can be used for real-time replication from SAP and also non-SAP sources.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

4. 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

5. Which of the following file types can be loaded using the flat file import wizard of Web IDE?

   Choose the correct answers.

   - [ ] **A** CSV
   - [ ] **B** TXT
   - [ ] **C** XLS
   - [ ] **D** XLSX

Learning Assessment - Answers

1. In SAP HANA, cold data is stored in HANA Native Storage Extension (NSE).

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct — Native Storage Extension is used to manage warm data. Cold data is stored outside of SAP HANA perhaps in Hadoop or SAP IQ.

2. What are typical characteristics of SAP HANA data replication?

   Choose the correct answers.

   ☐ **A** Virtualization

   ☒ **B** Duplicate data

   ☒ **C** Real-time

   ☐ **D** Continual streaming

   ☐ **E** Significant transformations

   Correct — Data replication with SAP HANA is typically the duplication of data in real time with no changes to data.

3. SAP SLT can be used for real-time replication from SAP and also non-SAP sources.

   Determine whether this statement is true or false.

   ☒ True

   ☐ False

   Correct — You can use SAP SLT for real-time replication from SAP and also non-SAP sources.

4. 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

   Correct — All features are available except SDA does not support cleansing or merging of multiple data sources.

5. Which of the following file types can be loaded using the flat file import wizard of Web IDE?

   Choose the correct answers.

   - [X] A CSV
   - [ ] B TXT
   - [X] C XLS
   - [X] D XLSX

   Correct — You can load CSV, XLS and XLSX but not TXT files using the flat file import wizard of Web IDE.

UNIT 5 Powering Data Warehouses with SAP HANA

UNIT OBJECTIVES

- Outline how SAP BW leverages SAP HANA
- Describe SAP HANA Data Warehouse Foundation
- Describe SAP Data Warehouse Cloud

Unit 5
Lesson 1

Running SAP Business Warehouse on 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

When SAP Business Warehouse (BW) was released in 1998, customers were able to choose their preferred database to run their SAP BW. In 2012, just two years after SAP HANA was released, SAP HANA was added to the list of databases that the customer could choose from to run their SAP BW.

Compared to the traditional disk based databases, SAP HANA provided significant improvements to the performance of SAP BW especially in the areas of reporting response time and data loading time. This was largely due to the fast, in-memory processing of SAP HANA. In addition to improved performance, SAP HANA also opens up new, powerful data modeling possibilities when customers combine SAP BW models with core SAP HANA models.

Almost immediately, SAP HANA became a very popular choice of database for new SAP BW customers and for the thousands of existing SAP BW customers who used the SAP-provided migration tools to migrate their SAP BW to run on SAP HANA.

SAP BW continued to support traditional non-SAP databases but the most significant new features that were added to SAP BW were only relevant for customers who were running their SAP BW on SAP HANA. The official name of the version of SAP BW that was is able to run on SAP HANA, in addition to legacy disk based databases, is called **SAP BW, powered by SAP HANA**.

Note:
SAP BW, powered by SAP HANA is an enhancement to the SAP BW code-line that goes back many years. SAP needed to ensure that this release of SAP BW could support all databases, legacy and SAP HANA. You will learn later about the new, leaner version of SAP BW that was completely re-written with a new code-line to support only SAP HANA.

SAP BW Simplified when Running on SAP HANA

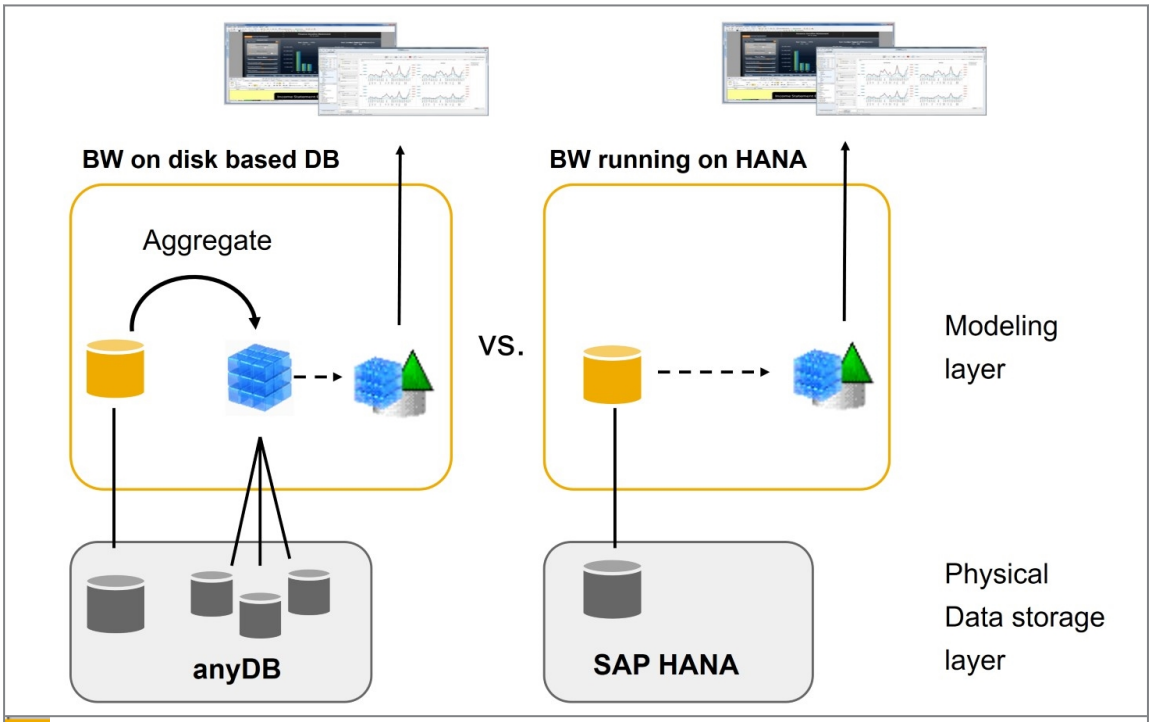


Figure 100: BW 1

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 due to the efficient storage technology of column store tables that massively compress data.

A non-HANA SAP BW design involves building many layers of pre-calculated and stored aggregations. We do this in order to improve the speed of reporting by pre-calculating business data in advance of the users requesting it. This is an extremely inefficient practice as we are storing multiple copies of data at various aggregation levels in the hope that users actually request data at the level of aggregations that we store. It also means that each aggregation needs to be continually refreshed as new data is loaded to SAP BW. With the raw power of SAP HANA, aggregations are done on the fly from the raw data and only when the user runs a query, so you no longer need to calculate and store aggregates.

By removing the stored aggregates, which are usually modeled using SAP BW InfoCubes and additional aggregates on top of 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 to calculate and load additional aggregates. Plus, it is easier to implement new data flows and integrate them into existing models when you only need to manage one layer of data and do not have to consider extra layers of aggregation.

With SAP BW, powered by SAP HANA, the only persistent layer is modeled with Advanced DataStore Objects (ADSO). These are only available with SAP BW powered by SAP HANA and are optimized for in-memory processing. On top of the Advanced DataStore Objects (ADSO) we model CompositeProviders which provide the virtual layer so that we can create various shapes of data for business consumption. CompositeProviders support the union and join of the underlying data sources and also additional calculation of new columns. CompositeProviders do not store data but calculate results on the fly. They are also optimized for in-memory processing of SAP HANA.

Integrate Any Data with BW

SAP HANA manages all SAP BW data in a single, dedicated database schema that is owned by SAP BW. But in addition to the BW-owned schema it is possible to create additional database schemas to manage data from other SAP and non-SAP applications.

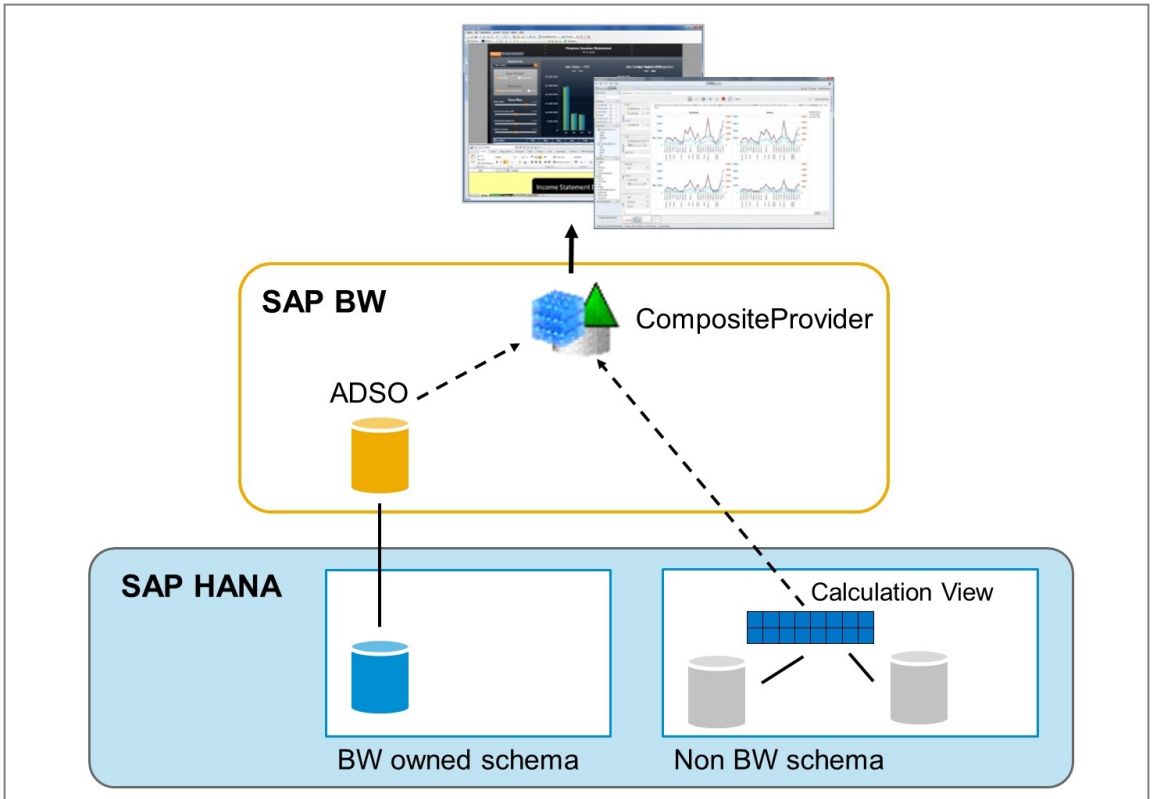


Figure 101: Integrate Any Data with SAP BW

SAP BW can access the tables and views of these additional SAP HANA schemas. You can leverage the advanced data acquisition tools that are part of SAP HANA, such as SDA, SDI, SLT, and streaming, to provision data to the tables of external schemas. You would then build SAP HANA calculation views on top of the tables in the external schemas, and finally integrate these calculation views with SAP BW modeled objects, such as CompositeProviders to create a mixed model.

This means that you can combine data warehouse data, perhaps representing the historical data, with SAP HANA data from external schemas representing live operational data. This is how SAP BW and SAP S/4HANA can provide a view that combines historical and current data. SAP BW reports and applications can then consume the hybrid model, giving you a complete picture of historical and up-to-date data for your real-time applications. This combination of the modeling features of SAP BW and SAP HANA is called mixed modeling or hybrid modeling.

Consume SAP BW Data from Any Application

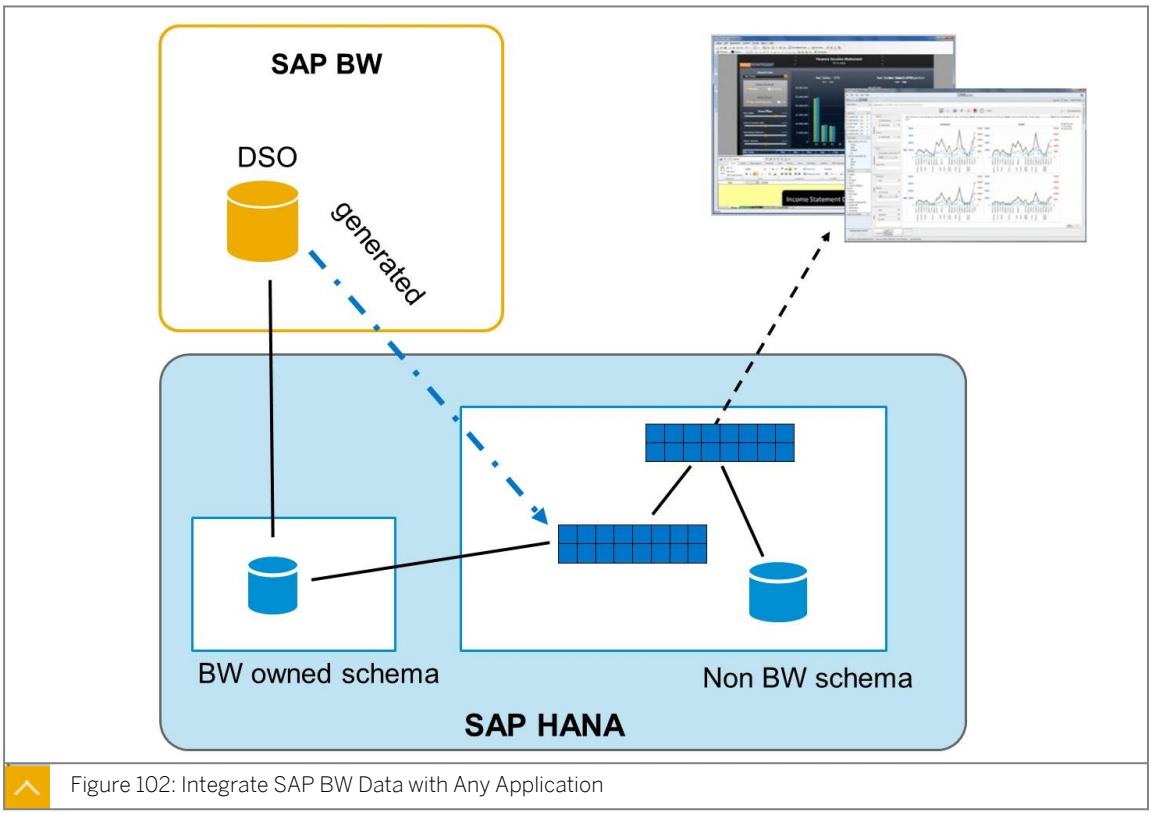


Figure 102: Integrate SAP BW Data with Any Application

SAP BW, powered by SAP HANA can automatically generate SAP HANA calculation views that correspond to SAP BW models such as InfoCubes, DataStore Objects, and CompositeProviders. You can then integrate these automatically generated SAP HANA calculation views with other SAP HANA calculation views that might provide additional calculations, to create a comprehensive, hybrid model that can be consumed by any tools. This also includes re-consuming the calculation view back into 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 sync.

SAP BW Powered by SAP HANA Versus SAP BW/4HANA

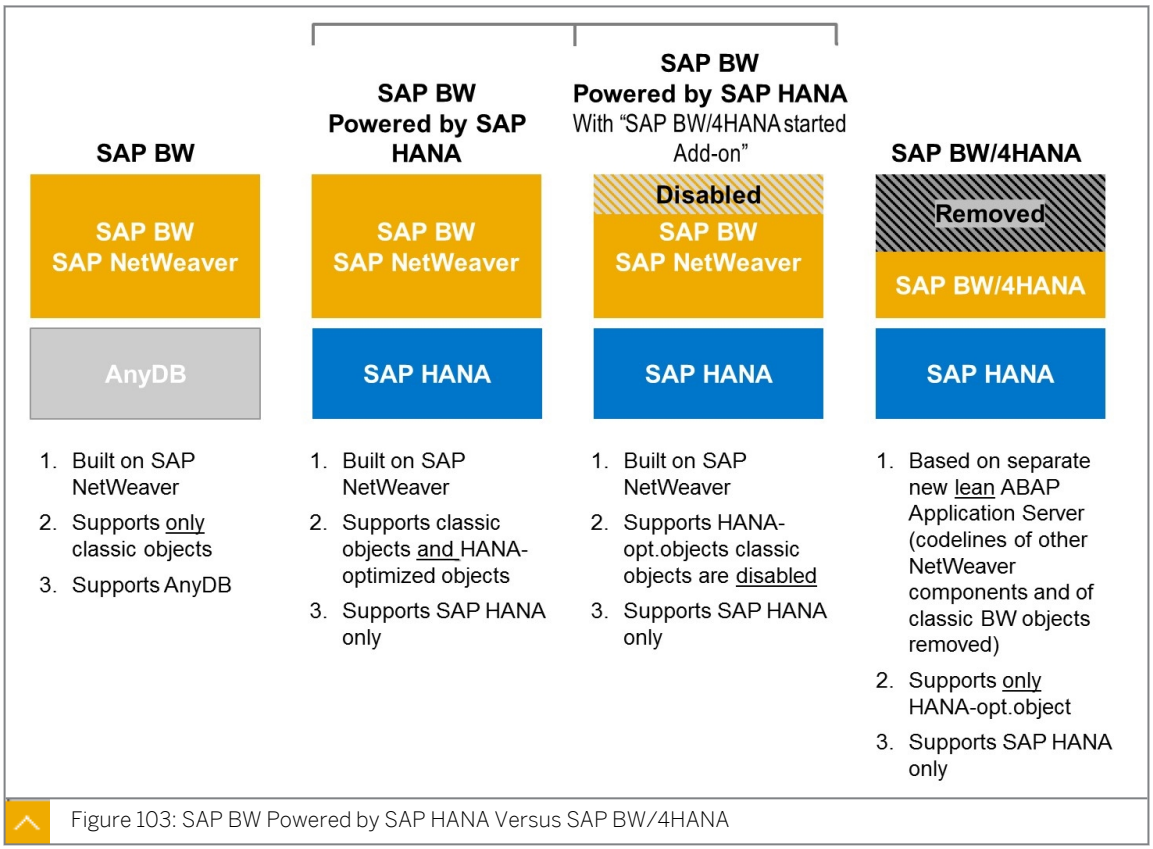


Figure 103: SAP BW Powered by SAP HANA Versus SAP BW/4HANA

There are two versions of SAP BW that can run on SAP HANA. They are as follows:

- SAP BW, powered by SAP HANA

  This is supported for SAP BW 7.4 and SAP BW 7.5, and is popular with customers who are transitioning from SAP BW on any DB. 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 SAP BW powered by SAP HANA, SAP developed an special add-in. Its purpose is to lock the legacy / classic objects of SAP BW for further development and maintenance. 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 SAP 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 the next-generation SAP 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. Legcay modeling objects such as InfoCubes and MultiProviders are no longer supported.

BEx Query Designer is not used. Queries are built using the Eclipse tools (specifically an Eclipse plug-in called BWMT) and SAPGUI is not needed for modeling or query building.

Customers can upgrade from BW powered by SAP HANA to SAP 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. The current version of SAP BW/4HANA is 2.0

You can compare BW powered by SAP HANA to Business Suite powered by SAP HANA, and SAP BW/4HANA to SAP 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.

Caution:
SAP BW powered by SAP HANA is often referred to as SAP BW on HANA, but this is not the official title. Also, SAP BW/4HANA is often confused with SAP BW powered by SAP HANA. Make sure you are clear that these are different products.

LESSON SUMMARY
You should now be able to:

- Outline how SAP BW leverages SAP HANA

Developing Custom SQL Data Warehouses with SAP HANA

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe SAP HANA Data Warehouse Foundation

Building a Custom Data Warehouse 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, SAP BW/4HANA. Both of these remain SAP's application (out-of-the-box)-driven approach for enterprise-wide data warehousing. However, SAP HANA offers an alternative approach, which does not require SAP BW or SAP BW/4HANA.

All tools required to build a high-performance SQL data warehouse are provided with the SAP HANA platform. This is referred to as the **Native SQL-driven data warehouse approach using SAP HANA**.

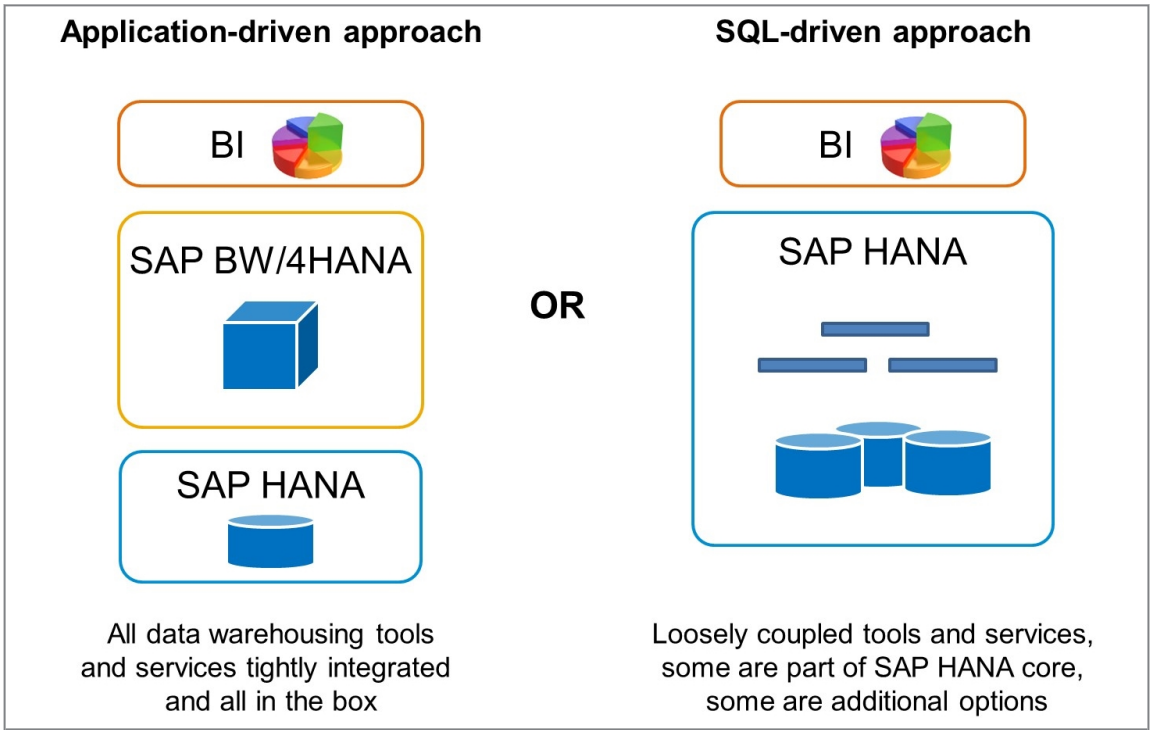


Figure 104: Choose a Data Warehouse Approach

With the application-driven approach, 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. Another aspect to consider is that SAP BW and SAP BW/4HANA are tightly integrated in to SAP enterprise suites and many other SAP applications, and that makes them appealing to customers who have already implemented SAP applications.

But some customers prefer to build the data warehouse themselves, using tools supplied by SAP. SAP HANA already includes most of the tools needed to build a data warehouse.

A data warehouse built using only SAP HANA is called an **SQL-driven data warehouse**. This name emphasizes that SQL development skills are key to the project. In addition to the provided tools with SAP HANA, SAP also provide dedicated tools for building a data warehouse from scratch. The benefit to this approach versus the application-driven approach is that customers can precisely define the architecture that suits them, starting small and growing into a full-scale enterprise-wide data warehouse. Compared to the application drive approach, the SQL-driven approach encourages a more 'open' data warehouse where all data can be freely accessed by any tools outside the platform. Also, a freestyle SQL programming approach allows a developer to go beyond what the standard application tools would provide.

Of course, there are costs to these benefits. The most obvious ones are the requirements for skilled developers and also the time, effort, and cost required to build a data warehouse from scratch.

Many customers have already begun their SQL-driven data warehouse journey. Also, customers have realized that both approaches complement each other where an application-driven approach is mixed with an SQL-driven approach to provide the optimal solution.

SAP provide a toolkit containing components that are specifically used to build a data warehouse. It is called **SAP HANA Data Warehousing Foundation**. it is developed by SAP and has its own release cycle. You use these additional tools in conjunction with the standard tools provided with SAP HANA to create sophisticated data warehouse.

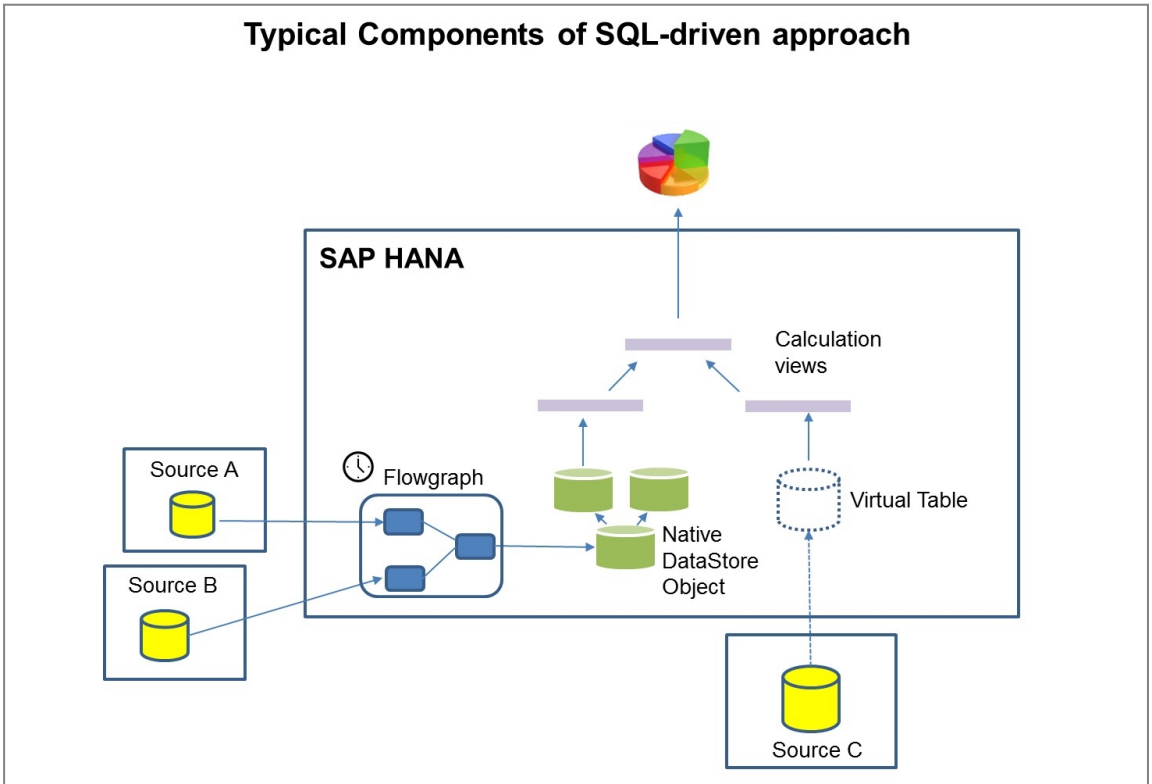


Figure 105: Components of SQL Data Warehouse

The following list outlines the key component that can be found in an SQL driven approach to data warehousing:

- Native DataStore Objects (NDSO): These are special native SAP HANA objects that capture and store data and provide a sophisticated delta mechanism for staging changes to data (compare them to SAP BW ADSO). They are provided as part of the SAP HANA Data Warehousing Foundation toolkit.
- Flowgraphs: Used to define the ETL data flow, using transforms that integrate, harmonize and cleanse difference data sources. This is a component of SAP HANA Smart Data Integration (SDI).
- Data Warehouse Scheduler (DWS): A component of the SAP HANA Data Warehouse Foundation toolkit that is used to schedule the flowgraphs to implement data load automation.
- Virtual Tables: A type of table that is permanently connected to a remote table so that live data is consumed. It does not store data. These are part of the standard SAP HANA database and use the Smart Data Access (SDA) mechanism.
- Calculation Views: Used to build the virtual layer. Highly re-usable and part of standard SAP HANA.

In addition to the two Data Warehousing Foundation components mentioned in the list above, there is also:

- Data Lifecycle Management (DLM) : A tool to manage the relocation of aged data to other remote sources such as warm store (disk-based storage) and cool store (data lakes), and vice versa. Enables SAP HANA administrators to model aging rules.
- 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. With DDO, you create predefined relocation rules. These rules define the conditions under which data should be moved and to where the data should be relocated.

LESSON SUMMARY

You should now be able to:

- Describe SAP HANA Data Warehouse Foundation

Unit 5
Lesson 3

SAP Data Warehouse Cloud

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

- Describe SAP Data Warehouse Cloud

SAP Data Warehouse Cloud

SAP provide three solutions for data warehousing:

- SAP BW/4HANA : a ready-made, on-premise data warehouse application.
- SAP SQL Data Warehousing : on-premise, tools to build-your-own custom SQL data warehouse.
- SAP Data Warehouse Cloud: cloud-based data warehouse.

SAP Data Warehouse Cloud is a pay-as-you-consume data warehouse hosted by SAP.

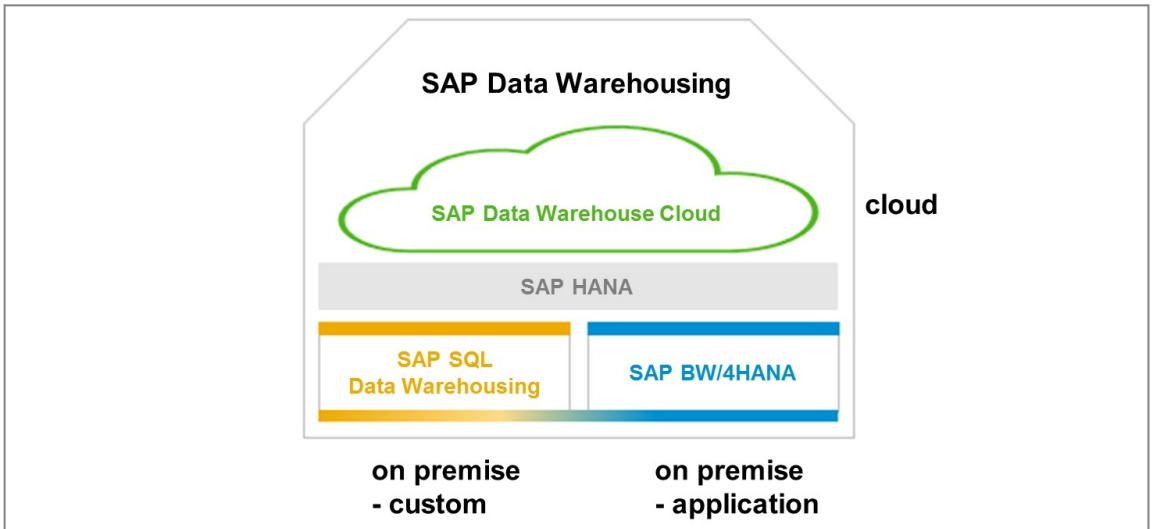


Figure 106: Three_DW_Solutions.pptx

SAP Data Warehouse Cloud is aimed at the business user and does not requires a significant IT operation. SAP run the solution ensuring it is always available and up to date, and of course always backed up. SAP Data Warehouse Cloud enables Lines of Business (LoB) to get started quickly using their own data sources, either local or enterprise, building their own data and business layers ready for consumption.

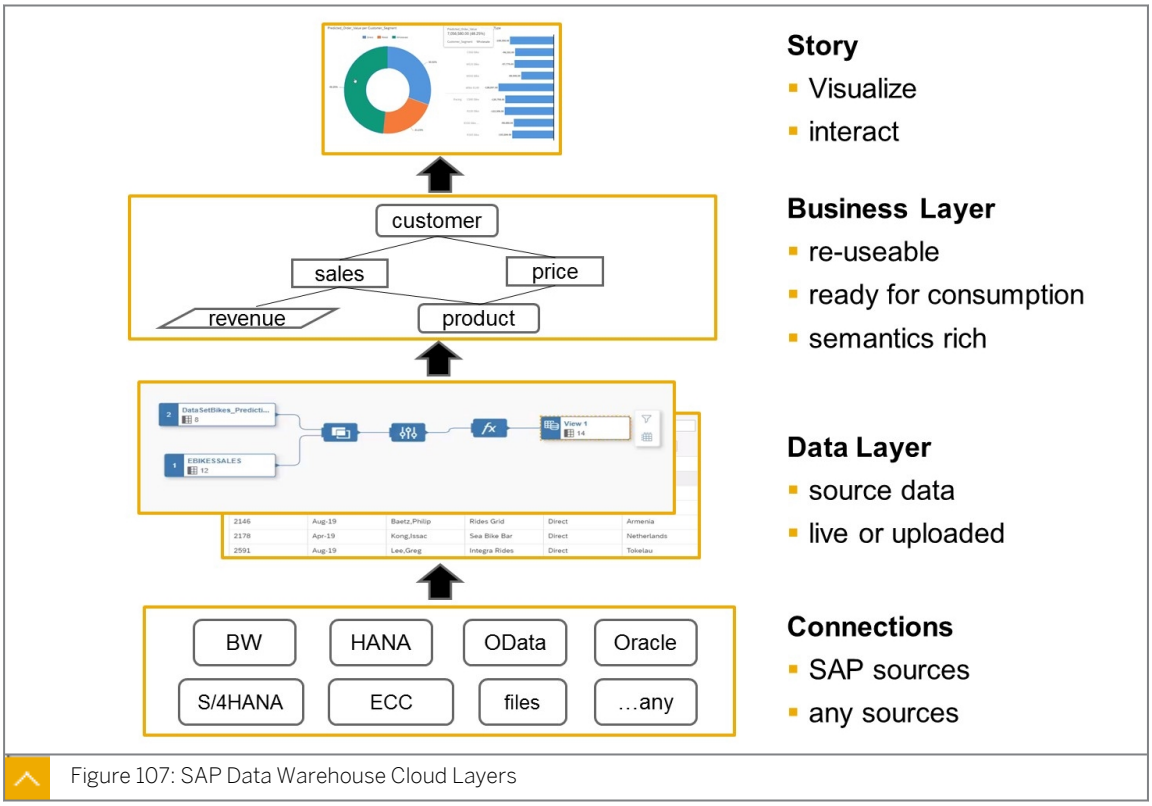


Figure 107: SAP Data Warehouse Cloud Layers

SAP Data Warehouse Cloud requires very little administration. A tenant can deployed in just minutes and you can choose the size (memory and disk) that you need to fit your requirements. If you need more or less resources later you can easily scale up or down and you pay for only what you need. This helps organizations to get started quickly, especially when they are not sure how well adopted their solution will be with the business and want to trial some scenarios.

SAP Data Warehouse Cloud sits in the middle of a stack which is made up of SAP HANA as the **data management layer**, then SAP Data Warehouse Cloud as the **data modeling layer**and then SAP Analytics Cloud as the **visualization layer**. SAP provision and connect all three components so the layers are invisible to the users. They use a very friendly, user-like, interface that guides them through all the setup and development steps.

SAP Data Warehouse Cloud also provides an optional integrated data lake which enables the reach of huge amounts of big data at low cost.

SAP Data Warehouse Cloud is not a replacement for SAP BW/4HANA or SAP SQL Data Warehousing. It has a unique use case. SAP BW/4HANA is an enterprise data warehouse that provides very advanced ETL and data modeling capabilities. It is important to remember that SAP BW/4HANA is an on-premise data warehouse and customers who want to maintain their own system with control over the administration, updates etc. might prefer this option.

SAP Data Warehouse Cloud can be seen as an extension to SAP BW/4HANA to allow lines of business to take charge of their own modeling and analytics using their own data.

LESSON SUMMARY

You should now be able to:

- Describe SAP Data Warehouse Cloud

Learning Assessment

1. In which key areas can we expect improvements when running SAP BW powered by SAP HANA?

   Choose the correct answers.

   - ☐ **A** Tighter security
   - ☐ **B** Data loading performance
   - ☐ **C** Superior data quality
   - ☐ **D** Reporting performance

2. Why do I install SAP HANA Data Warehousing Foundation add-on?

   Choose the correct answer.

   - ☐ **A** To enhance the performance of SAP BW powered by SAP HANA.
   - ☐ **B** To provide the essential components needed to develop an SQL-driven custom data warehouse.
   - ☐ **C** To move my SAP BW/4HANA on-premise solution to the cloud.

3. What are the valid architecture layers of the SAP Data Warehouse Cloud?

   Choose the correct answers.

   - ☐ **A** Business Layer
   - ☐ **B** Aggregation Layer
   - ☐ **C** Connections
   - ☐ **D** Data Layer

Learning Assessment - Answers

1. In which key areas can we expect improvements when running SAP BW powered by SAP HANA?

   Choose the correct answers.

   - [ ] A Tighter security
   - [x] B Data loading performance
   - [ ] C Superior data quality
   - [x] D Reporting performance

   Correct — Security and data quality are not improved when running BW powered by SAP HANA but loading and reporting performance are significantly improved.

2. Why do I install SAP HANA Data Warehousing Foundation add-on?

   Choose the correct answer.

   - [ ] A To enhance the performance of SAP BW powered by SAP HANA.
   - [x] B To provide the essential components needed to develop an SQL-driven custom data warehouse.
   - [ ] C To move my SAP BW/4HANA on-premise solution to the cloud.

   Correct — SAP HANA Data Warehousing Foundation add-on provides the essential components needed to develop an SQL-driven custom data warehouse.

3. What are the valid architecture layers of the SAP Data Warehouse Cloud?

   Choose the correct answers.

   - [x] A Business Layer
   - [ ] B Aggregation Layer
   - [x] C Connections
   - [x] D Data Layer

   Correct! From the bottom, the valid layers are Connections > Data Layer > Business Layer > Visualization Layer (not mentioned in answers).

UNIT 6 Running SAP Enterprise Suites on SAP HANA

Lesson 1

UNIT OBJECTIVES

- Describe how enterprise suites run on SAP HANA

Unit 6
Lesson 1

Running SAP Enterprise Suites on SAP HANA

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

- Describe how enterprise suites run 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. 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.

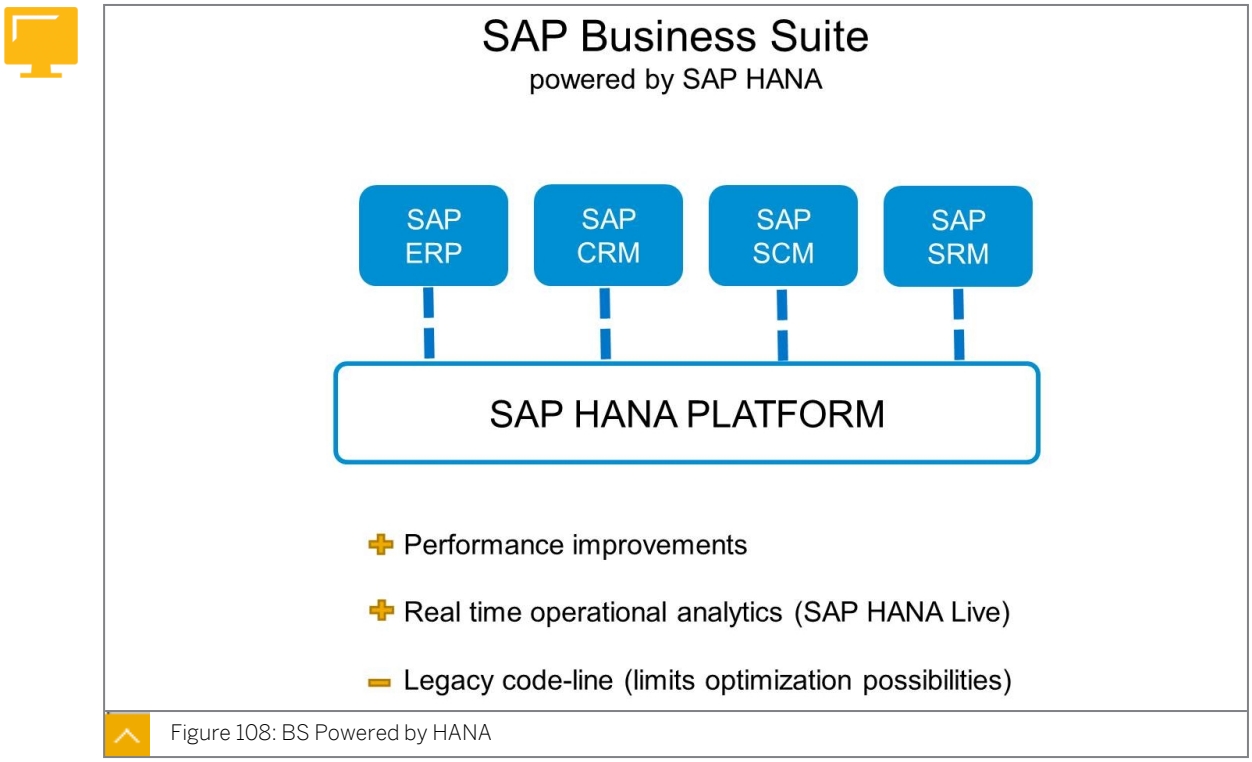


Figure 108: BS Powered by HANA

SAP Business Suite powered by SAP HANA was born and provided immediate benefits. The following list outlines 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, remember 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, and so on). 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 has 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

What is SAP S/4HANA?

SAP S/4HANA (Suite for HANA) is SAP's next-generation business suite for the digital world. SAP S/4HANA is built and optimized to run only on the SAP HANA database. It is not available on any other third party database such as Oracle, IBM, or Microsoft.

SAP S/4HANA includes solutions for all key business processes such as procure-to-pay, order–to-cash, and so on.

Many customers have migrated from Business Suite on anyDB to Business Suite powered by SAP HANA to obtain a massive speed-up in performance of their business applications. The move to SAP S/4HANA is often the first major step for a customer who is building a next-generation business platform.

For customers who did not migrate from Business Suite on anyDB to Business Suite powered by 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 simplify and shorten the data migration effort.

SAP S/4HANA was completely rewritten from scratch. Just like Business Suite, SAP S/4HANA is still written in ABAP, but the ABAP code was 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 the code was adjusted to ensure that it could run on SAP HANA with some optimizations added.

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 fitted the technology of that time. There were many approaches used to ensure best performance on legacy databases that are simply not needed today with SAP HANA. For example, SAP S/4HANA does not need to store aggregated data. The SAP HANA database used by SAP S/4HANA 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 has to support multiple databases, so the code became complex to ensure compatibility with all of them. SAP S/4HANA only runs on SAP HANA, and so we could drop

the unnecessary code and produce a leaner, faster, more efficient ERP suite that is ready to move companies forward with a foundation for their digital transformation.

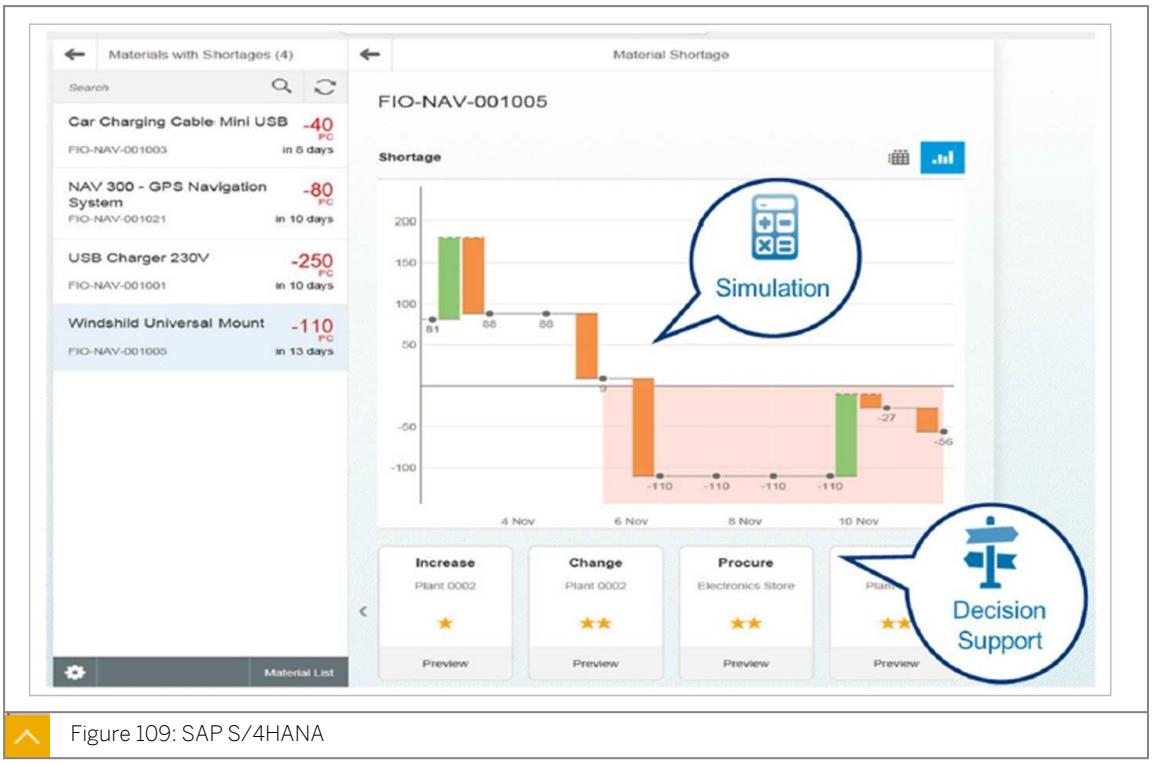


Figure 109: SAP S/4HANA

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.

A major difference between Business Suite and SAP S/4HANA is the provision of many brand-new applications. Business Suite powered by SAP HANA focuses on continuity. SAP S/4HANA provides continuity, while also providing next-generation digital applications that combine in-memory analytics and transactional processing. SAP has rebuilt many traditional applications so that they provide the business user with analytics, right in place where decisions are needed.

Embedded Analytics

A key component delivered with SAP S/4HANA is Embedded Analytics.

Embedded Analytics provides a complete enterprise-wide data model, built using ABAP Core Data Services (CDS). The data model delivers thousands of consumption-ready CDS views of real-time business data. SAP S/4HANA Embedded Analytics also includes a built-in analytic engine (based on BW) to handle complex OLAP requests in memory. Also supplied are built-in, easy-to-use query building tools. It is possible to use any existing SAP reporting tool, such as Analysis, Web Intelligence or Lumira with Embedded Analytics.

LESSON SUMMARY

You should now be able to:

- Describe how enterprise suites run on SAP HANA

Learning Assessment

1. What are features of SAP Business Suite powered by SAP HANA?

   Choose the correct answers.

   - ☐ **A** Includes Embedded Analytics
   - ☐ **B** ABAP is replaced with JAVA
   - ☐ **C** Combine operational analytics and transaction processing
   - ☐ **D** Performance improvement

2. What are features of SAP S/4HANA?

   Choose the correct answers.

   - ☐ **A** Optimized for SAP HANA but maintains compatibility with other databases
   - ☐ **B** Embedded Analytics
   - ☐ **C** Rewritten ABAP code optimized for SAP HANA
   - ☐ **D** SAP HANA Live included to provide real time operational analytics

Learning Assessment - Answers

1. What are features of SAP Business Suite powered by SAP HANA?

   Choose the correct answers.

   - [ ] A Includes Embedded Analytics
   - [ ] B ABAP is replaced with JAVA
   - [X] C Combine operational analytics and transaction processing
   - [X] D Performance improvement

   Correct — Embedded Analytics are only relevant for SAP S/4HANA. Business Suite powered by SAP HANA uses SAP HANA Live. ABAP is still used. Business Suite powered by SAP HANA now combines operational analytics and transaction processing and you get a boost in performance.

2. What are features of SAP S/4HANA?

   Choose the correct answers.

   - [ ] A Optimized for SAP HANA but maintains compatibility with other databases
   - [X] B Embedded Analytics
   - [X] C Rewritten ABAP code optimized for SAP HANA
   - [ ] D SAP HANA Live included to provide real time operational analytics

   Correct — SAP S/4HANA runs only on the SAP HANA database. 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.

UNIT 7 Developing Applications on SAP HANA

UNIT OBJECTIVES

- Understand the role of the ABAP developer in an SAP HANA Implementation
- Describe an SAP HANA Native Application

Developing ABAP applications for SAP HANA

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Understand the role of the ABAP developer in an SAP HANA Implementation

The Role of ABAP in an SAP HANA Implementation

ABAP is the development language of SAP enterprise applications. ABAP continues to play an important role in an SAP HANA implementation because SAP HANA powers many of the ABAP-based applications developed by SAP, partners and customers.

ABAP Applications - Native SAP HANA or Legacy?

In the context of SAP HANA, there are two main types of ABAP application: those that were built to run on any database, such as SAP Business Suite and SAP CRM, and those that were built to run only on SAP HANA, such as SAP S/4HANA and SAP C/4HANA.

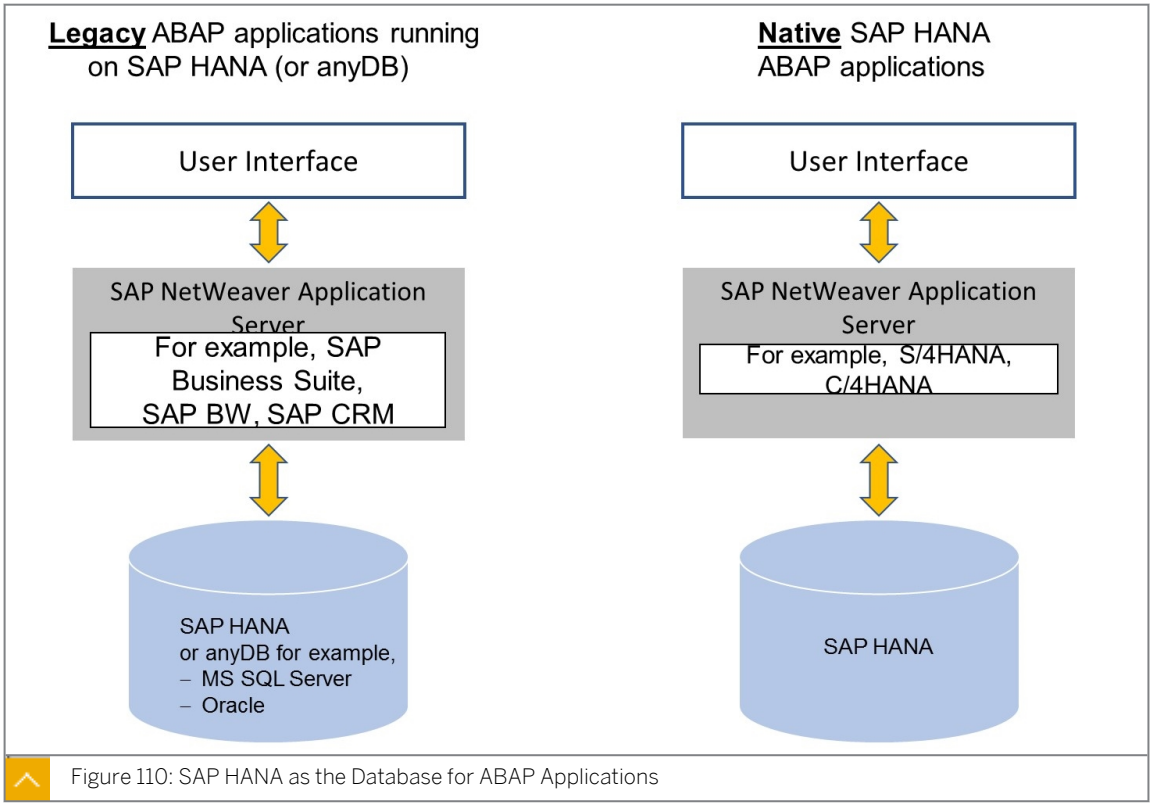


Figure 110: SAP HANA as the Database for ABAP Applications

Applications that were built to run an any database contain generic ABAP code that any database can interpret. And that also includes SAP HANA. This often means that the ABAP code misses the opportunity to exploit individual database-specific features that might improve application performance. Whilst some performance gains can be expected when

running these legacy applications on SAP HANA, due to the use of in-memory processing, SAP HANA performance is not fully exploited as the legacy, none-optimized, generic ABAP code is still being used.

Applications that were built to run only on SAP HANA contain code that exploits the specific features of the SAP HANA in-memory database to achieve optimal performance. These applications should not on any other database as the native code might not work well due to specific SAP HANA features being used in the code. These types of applications provide the best possible performance and exploit all optimization features of SAP HANA.

An ABAP developer who works on an application running on SAP HANA, whether it was originally built to run on any database or was built only for SAP HANA should always follow the best practices for developing optimal ABAP code to get the best from the performance of SAP HANA and they should develop skills to ensure they are able to do this.

Tools Used by ABAP Developers in SAP HANA

In the past, ABAP developers used SAPGUI as the main tool for code editing and testing. Although this tool can still be used in an SAP HANA development project, there is another, more modern tool that is recommended. The tool is referred to as ADT, which is an acronym for ABAP Development Tools. ADT is not really a tool but a plug-in developed by SAP that provides a highly productive interface for ABAP development and is built on the well known open source software development platform called Eclipse.

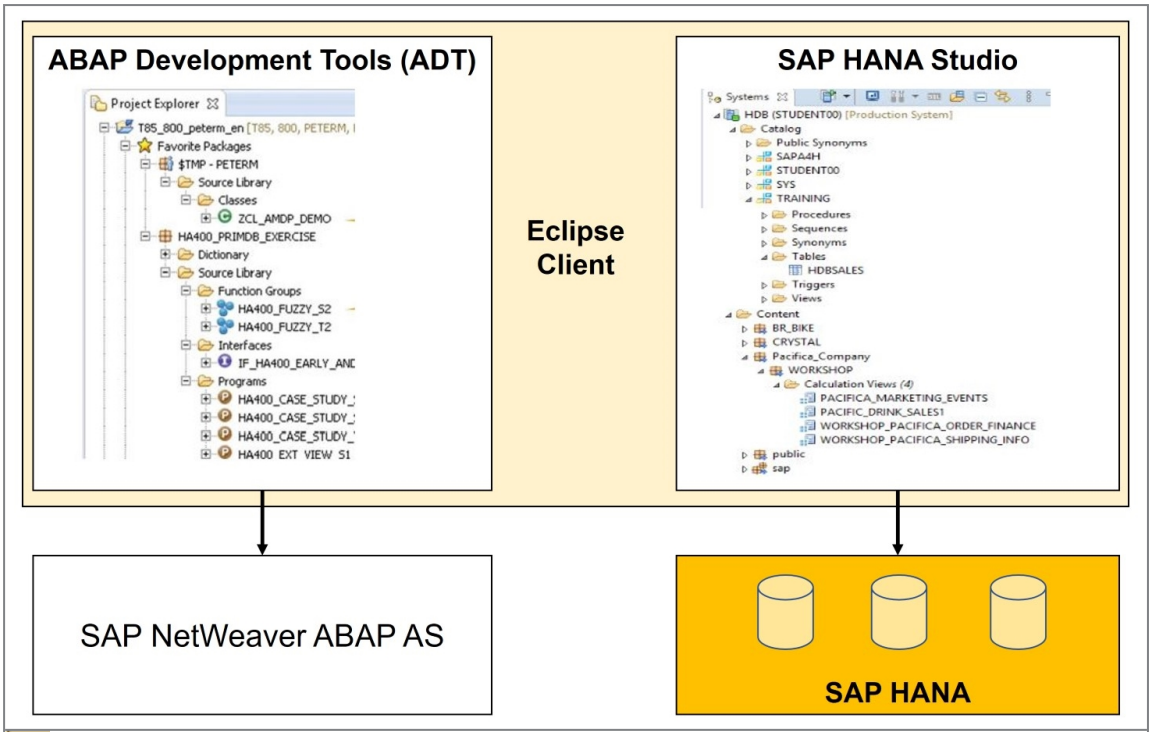


Figure 111: Tools for ABAP Developers in SAP HANA Projects

ABAP developers will recognize all objects and components in ADT such as packages, function modules, reports, and DDIC elements such as domains and tables. ADT contains many productivity tools to support the developers such as debuggers, object dependency tracing and test tools.

One of the key benefits of using ADT versus SAPGUI is that it is easy to switch between ABAP code development and the SAP HANA database. This is because access to the SAP HANA database is also through Eclipse. SAP have developed another plug-in called SAP HANA Studio that provides tools for the database administrator who is able to explore database

objects such tables, views and procedures. This view might also be of interest to the ABAP developer who would like to switch between the ABAP layer and the SAP HANA database layer. The developer simply opens Eclipse and easily switches between ADT and SAP HANA Studio views with a single click.

Opportunities for ABAP Developers in SAP HANA Projects

So what is the specific role of an ABAP developer in an SAP HANA project?

ABAP applications that run on SAP HANA are still ABAP applications. So we need ABAP developers to maintain and develop the enhancements to the applications. But an ABAP developer who will work on an application running on SAP HANA should learn how to code ABAP in a way that enables the best possible performance. Also, as many customers plan the migration of their current ABAP applications that run on legacy, disk-based databases to SAP HANA, the ABAP developer needs to be aware of the potential legacy code issues that could cause customer or partner-built ABAP extensions to run sub-optimally or even incorrectly on SAP HANA. They should know how to locate, adjust and test the troublesome code.

But there are also new opportunities too for the ABAP developer.

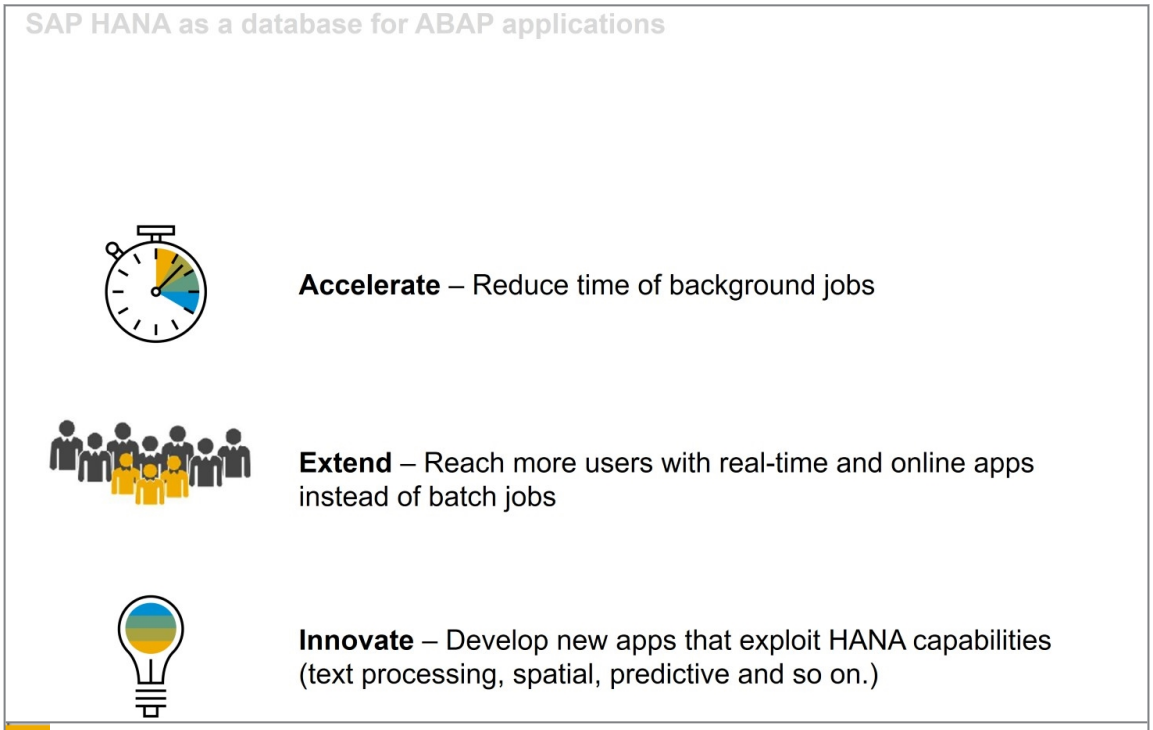


Figure 112: Opportunities for ABAP Developers in SAP HANA Project

The ABAP developer is responsible for evaluating code to identify how applications could run even faster when migrated to SAP HANA. Sometimes simple code tweaks can make a big difference to how an application performs.

An application that was once run in the background due to the time it took to execute, for example material requirements planning (MRP), might be possible to execute as an online application if run-time can be significantly reduced with code adjustments.

Developing ABAP applications on SAP HANA opens up many new opportunities that leverage the unique data processing capabilities of SAP HANA, such as text, spatial and predictive analytics. Even simple developments that embed native SAP HANA fuzzy text searching into applications are easy to implement and provide powerful features to end users.

Migration - Key Differences and Potential Functional Issues with ABAP Code

Large numbers of customers are keenly migrating their SAP, partner and home-grown ABAP applications to run on SAP HANA to benefit from improved performance and overall architecture simplification.

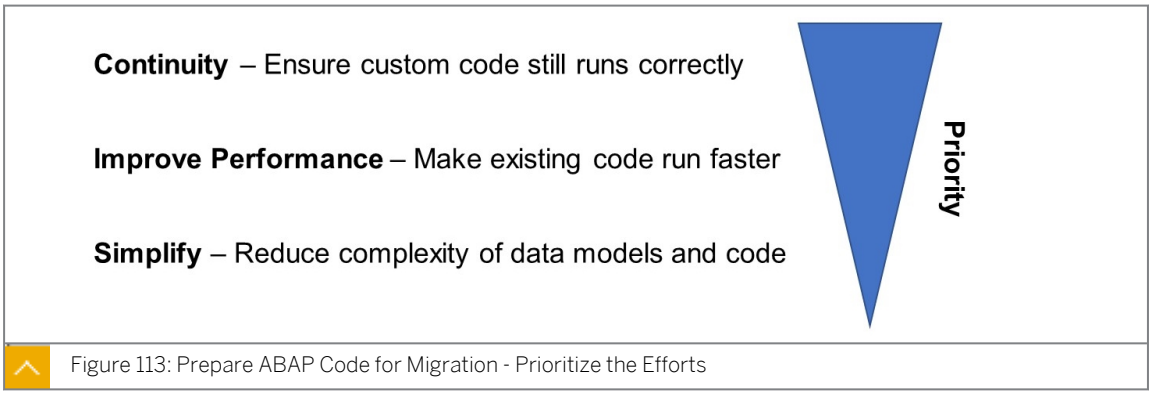


Figure 113: Prepare ABAP Code for Migration - Prioritize the Efforts

However, when migrating legacy applications to run on SAP HANA it is important to evaluate the ABAP code to ensure it still runs as expected on SAP HANA. This, of course is the highest priority. ABAP code that was originally created to run on any database might not work well, or even at all on SAP HANA. Code adjustments might be necessary. Developers will use various tools to locate and adjust the code so it runs optimally on SAP HANA.

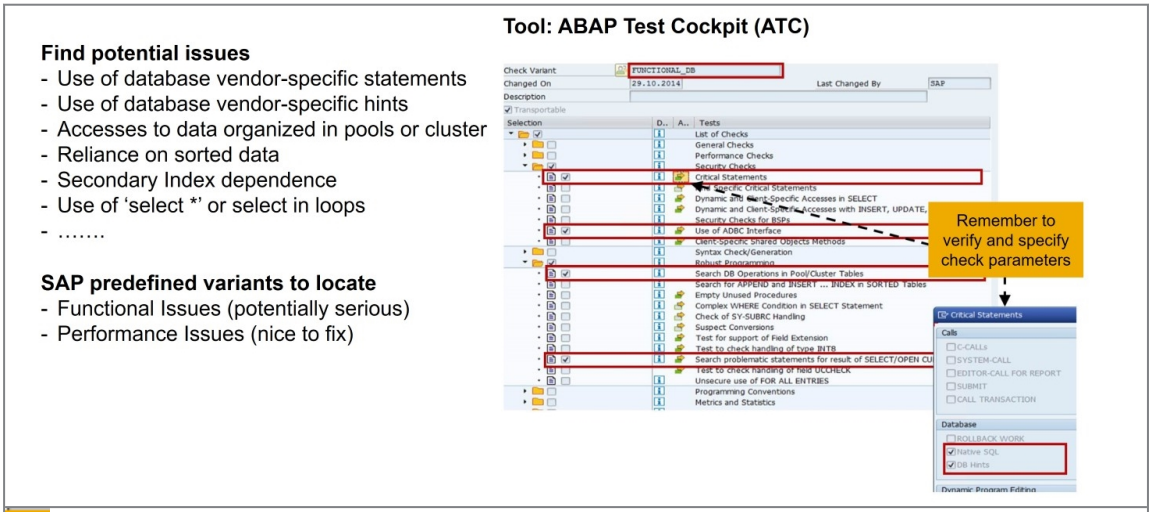


Figure 114: Differences and Potential Issues with ABAP Code

So what are the key differences in the ABAP code that developers should be aware of?

Legacy ABAP code might contain vendor-specific database code such as native SQL called by `EXEC SQL`. There might be embedded database hints that are vendor specific.

SAP HANA does not use pool or table clusters and only uses transparent tables, so code that explicitly references these should be evaluated.

Secondary Indexes are usually not required and so references to these in the ABAP code must be checked.

If an application requires data to be provided in a specific sorted sequence this must be checked because you should not rely on the implicit sort order of data returned by primary or secondary keys.

SAP provide a comprehensive tool to help locate and adjust the ABAP code that needs evaluating. It is called the ABAP Test Cockpit. This tool can be used on code that is going to be migrated so developers can assess the impact and calculate the effort for adjustment before committing to the migration. The tool not only locates the code of interest, but also provides workflow features around this, for example, to document the management decision regarding the action on the code.

Potential Performance Optimizations

Once the applications are running as expected, ABAP developers can look for opportunities to further improve performance with code adjustments. These adjustments can range from simple one line code tweaks to a complete redesign of whole chunks of code.

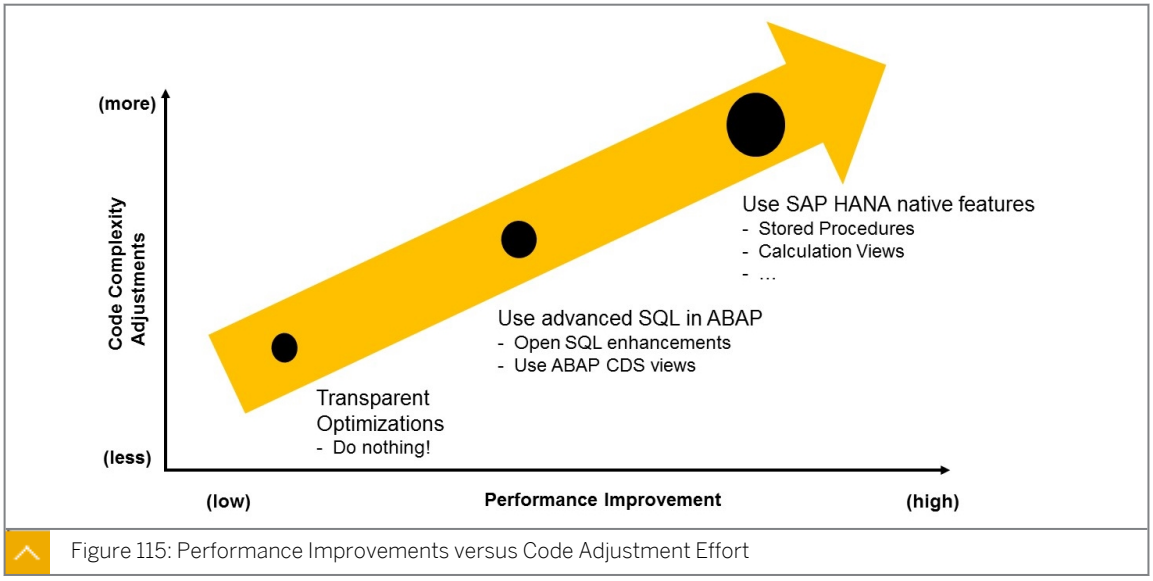


Figure 115: Performance Improvements versus Code Adjustment Effort

Migrating ABAP applications to run on SAP HANA yields performance improvements over disk based databases. Some of these improvements are immediately realized with no code adjustment due to the switch from disk-based technology to much faster, in-memory technology.

But often many more performance improvements can be realized with code adjustments or by making use of the latest ABAP features that are specifically optimized for SAP HANA.

One of the most basic differences is that the commonly used `SELECT *` , which means select all columns of a view of table, is not recommended. Developers should change these statements to select only the columns required by the application. This is because SAP HANA uses column store tables which organise data in columns and not rows. Selecting all columns in a column store table is very expensive, eats up memory and harms performance.

As well as `SELECT *` the developer should also look out for the use of selects within loops. Since it became possible to use joins in ABAP, the common practice of selecting records from one table whilst reading another was no longer recommended. However, many developers did not adjust their code as the gains were perhaps not so significant to justify the effort. But select within loops is now definitely not recommended for ABAP running on SAP HANA. A recode is recommended to get the best performance.

Often, the best performance gains can be realized by re-creating some of the ABAP development objects as SAP HANA database objects such as stored procedures and columns views. But this comes at the cost of additional development effort, plus not all ABAP developers will have the skills to develop native SAP HAN database objects.

Migrated applications bring along with them a lot of unwanted complexity not only in the code, but also in the underlying data model. ABAP developers have an opportunity to simplify the data models and therefore the code that runs on the model can also be simplified. For example, modern SAP HANA applications often do not need stored aggregates or additional indexes to assist with performance. These can be removed from the data model, saving storage and also reducing code complexity. Reducing model and code complexity means easier maintenance.

Push Down Processing to SAP HANA

Prior to SAP HANA, it was recommended to keep the processing load away from the database server and to utilise the ABAP server for the heavy processing of data. With SAP HANA this is now completely changed. SAP HANA is built for high-volume data processing. It processes data in-memory and not from disk. SAP HANA parallelizes data processing and uses sophisticated partitioning and caching. So, it is now recommended to push the data processing load to the SAP HANA database to achieve the best possible performance for an application.

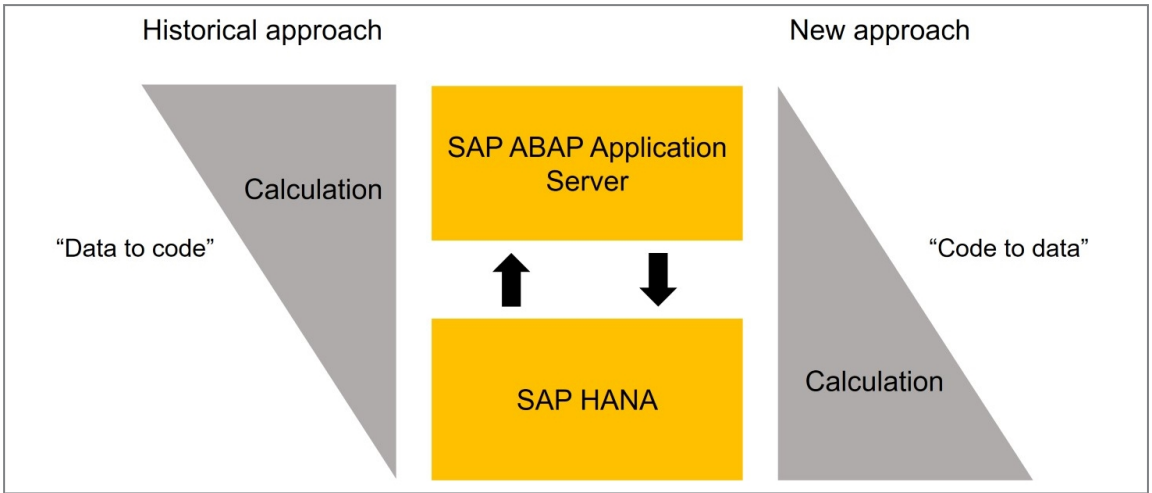


Figure 116: ABAP Code to Data

So if we are shifting processing to SAP HANA, is ABAP no longer required?

ABAP is most definitely required because although SAP HANA is a superior processor of data processing tasks such as aggregations, filters, ranking, currency conversion and so on, SAP HANA is not able to handle complex business logic or manage user interaction with the application through interfaces.

Note:
SAP HANA uses SQL as its data processing language and this is a declarative language which is ideal for issuing complex queries to the database. ABAP is an imperative language and is ideal for defining complex control logic (Do, Loop, If-then..). We need both languages in a modern development.

In simple terms, ABAP becomes the orchestrator, sending data processing requests to SAP HANA and then using its superior business logic handling capabilities it determines what happens next in the business process flow. The key thing is that ABAP should not be used to handle data processing tasks, these should be passed to SAP HANA.

By pushing data processing to SAP HANA we are avoiding sending all the raw data to the ABAP server for processing and then sending back the result to SAP HANA database. By pushing the code (the request) to SAP HANA, the data never leaves the database. All processing is done in the database where the data lives. ABAP is merely the conductor of this action.

Implementing Push-Down - Top Down or Bottom Up?

So how does the ABAP developer implement push-down to SAP HANA?

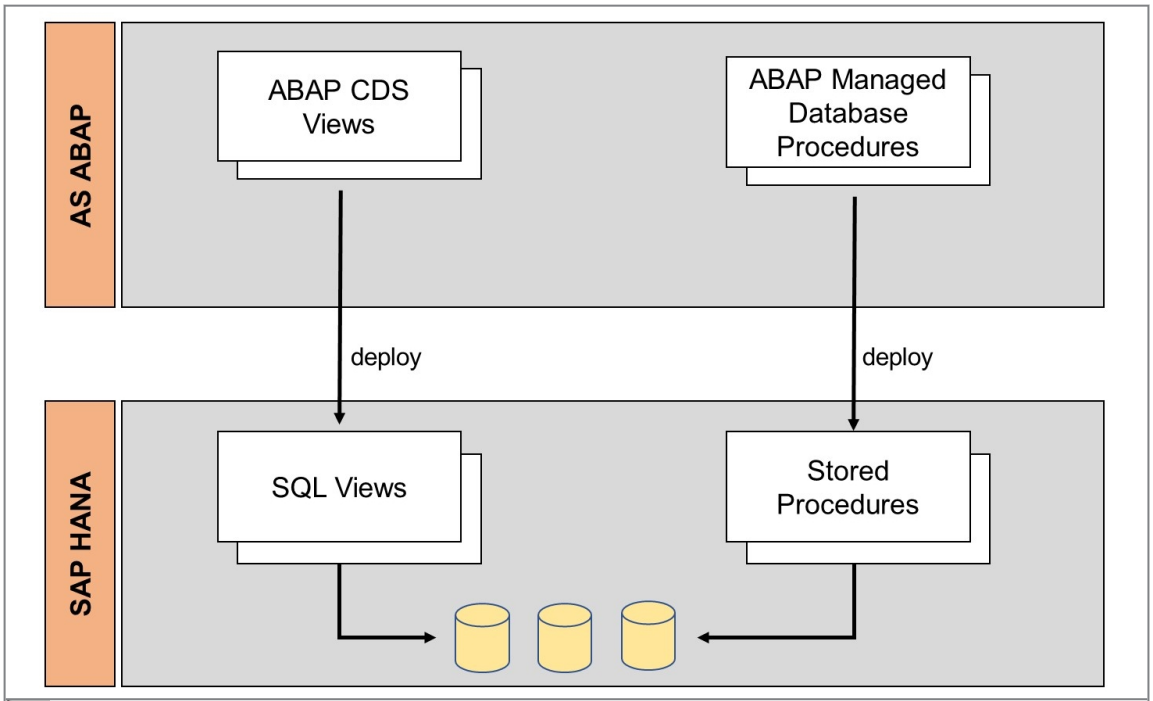


Figure 117: Top Down

There are two approaches: top down and bottom up.

The top down approach is implemented by developing ABAP objects in the ADT tool that generate native SAP HANA SQL objects such as views or stored procedures.

ABAP Core Data Services (CDS) is a data definition language developed by SAP and extends the traditional ABAP Data Dictionary to include powerful, semantically-rich views. ABAP CDS views are developed as ABAP objects in the ADT tool and are transported with other ABAP objects. When they are activated they generate SQL views in the SAP HANA database. The generated views should not be consumed directly or maintained directly in SAP HANA. They are owned and managed by ABAP CDS.

Caution:
Do not confuse ABAP CDS with SAP HANA CDS. They are based on the same concept but are technically different.

ABAP Managed Database Procedures (AMDP) are re-useable procedures written in one of the native database languages of SAP HANA, which is called SQLScript. But unlike stored procedures developed directly in SAP HANA, AMDP's are built using the ADT tool in the ABAP layer and are wrapped up as an ABAP object and are managed and transported as part of an overall ABAP project.

So it is called the top down approach to push-down because all development takes places from the 'top' which in this case is the ABAP layer, and SAP HANA objects are always generated and never created directly. The ABAP developer does not need to work in the SAP HANA database and stays in the ABAP layer.

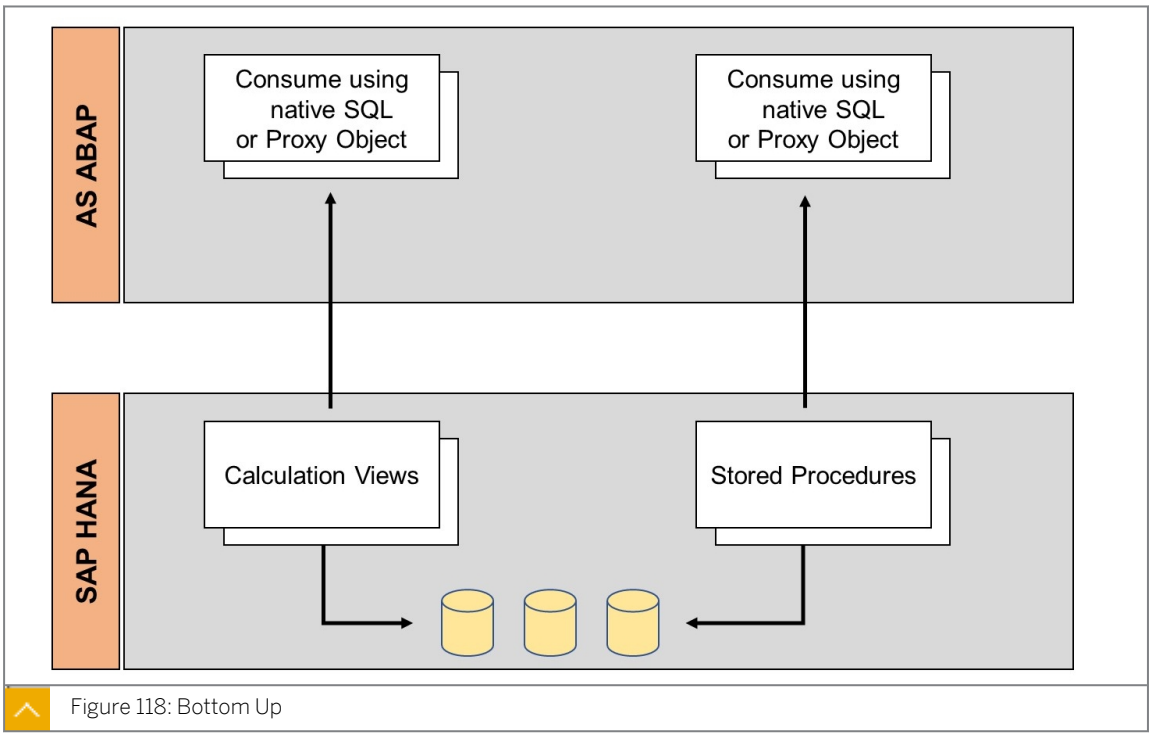


Figure 118: Bottom Up

In contrast, the bottom-up approach is when native SAP HANA re-useable database objects such as calculation views and stored procedures are developed in SAP HANA using SAP HANA Studio, and then consumed by ABAP code that is able to interact with the SAP HANA objects by passing parameters. The bottom up approach usually requires more effort to implement than the top down approach as we have to first develop native SAP HANA objects and then write ABAP code to consume those objects. Also it requires knowledge of column views and stored procedure development in SAP HANA. Also bear in mind, transports could be more complex as we build objects in two development environments. In the bottom up-approach, the ABAP developer works in both the ABAP layer and the SAP HANA database, or they rely on other team members to develop the SAP HANA database objects.

Note:
To learn more about ABAP development on SAP HANA, you may want to consider the course HA400, *ABAP Programming for SAP HANA*.

LESSON SUMMARY
You should now be able to:

- Understand the role of the ABAP developer in an SAP HANA Implementation

Unit 7
Lesson 2

Developing Native SAP HANA Applications

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

- Describe an SAP HANA Native Application

Introduction to Developing Native SAP HANA Applications

First, let's make sure we understand what we mean when we refer to **native** SAP HANA applications.

When we refer to any SAP HANA application, we need to differentiate between the two types of applications:

- Applications that run on their own (non-HANA) application server and use SAP HANA only for the database services.
- Applications that are built and run directly on SAP HANA and use the SAP HANA built-in application server as well as the SAP HANA database services. These applications require no separate application server.

The following list outlines some applications that require their own dedicated application servers:

- SAP S/4HANA
- SAP Business Suite
- SAP BW/4HANA

As these applications are built using ABAP, they need to run on SAP NetWeaver Application Server. SAP HANA is not an ABAP application server and this is why the additional server is needed to support these applications. In this scenario, SAP HANA provides the database services to the NetWeaver ABAP server. SAP HANA then becomes the bottom layer in a 3-tier stack made up of the GUI at the top, then the application server layer, and finally the database services layer at the bottom. In this case, the applications are not native to SAP HANA. They are either **powered by** SAP HANA, if they are applications that can run on both SAP HANA and legacy databases, or they are **4/HANA** if they were built just to run only on SAP HANA.

But an application that is built to run on the embedded application server of SAP HANA (XS Advanced), is referred to as a **native** SAP HANA application. In the native application scenario we need just 2 tiers, the GUI and SAP HANA.

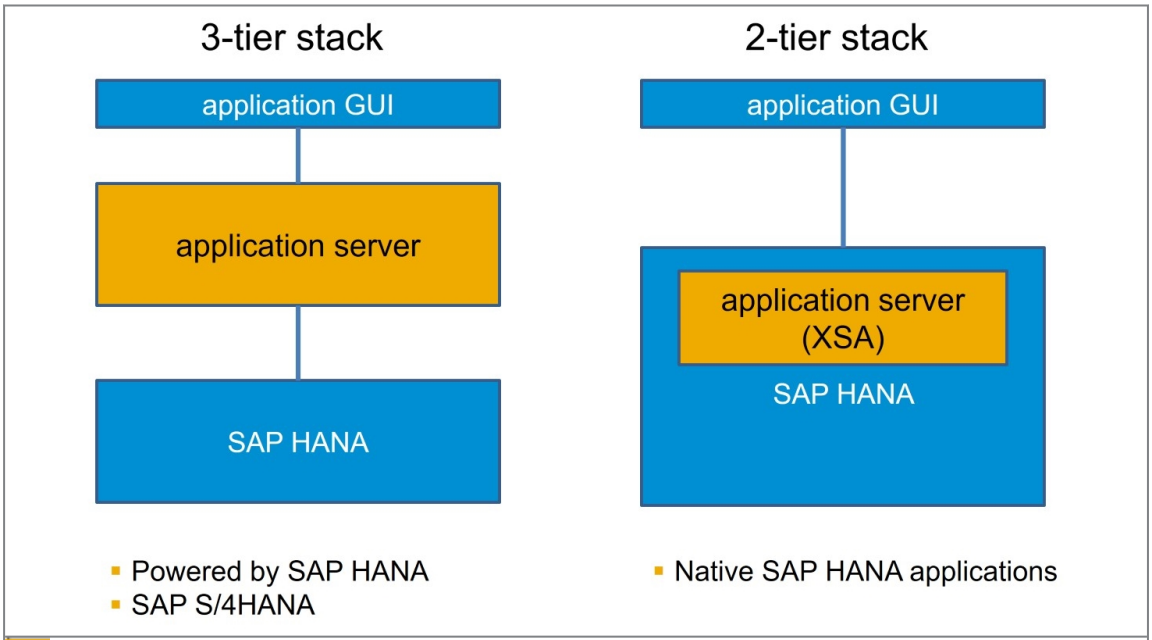


Figure 119: Native Versus Non–Native HANA Applications

Building Native Applications on SAP HANA

SAP HANA contains all the components required to develop and run applications in a simple two-tier stack. There is no need to include an additional application server and this means less hardware and less complexity, and, of course, better performance without the connectivity between servers.

SAP HANA supports modern development languages and includes a full development and run-time environment for all types of applications. Any type of application can be built from mobile applications, corporate dashboards to enterprise-wide applications.

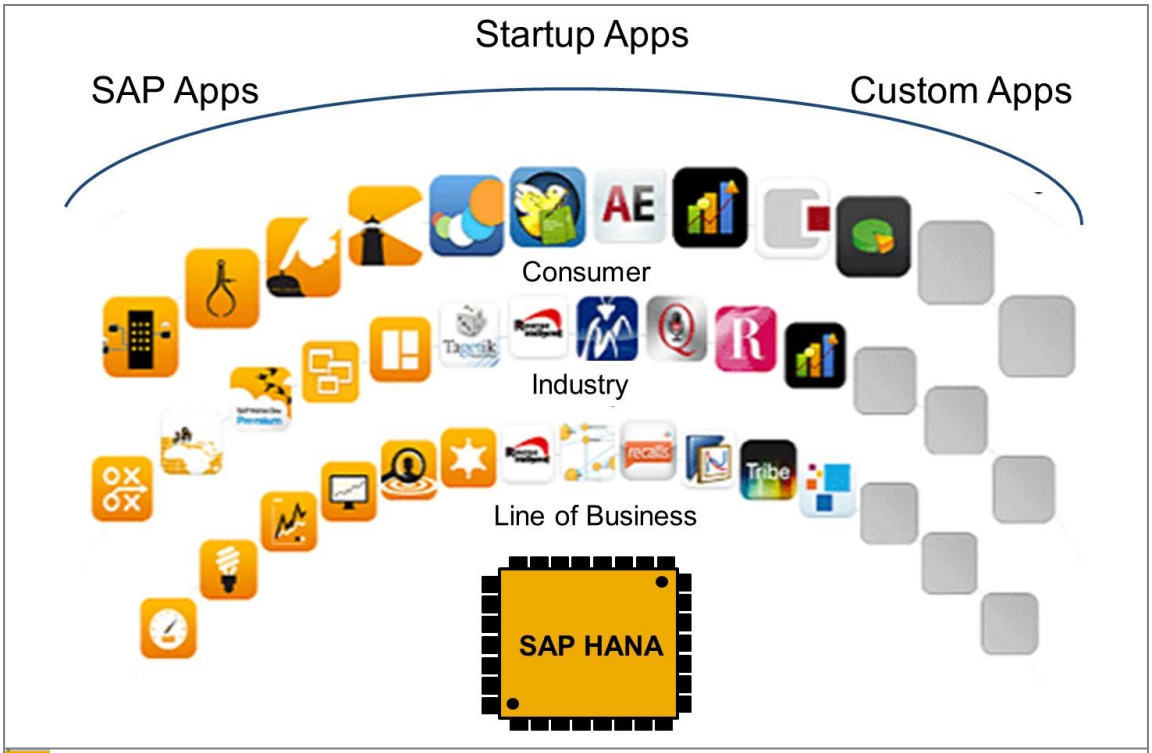


Figure 120: Native Applications on SAP HANA

With SAP HANA, you can build **scalable** applications that start small with minimal infrastructure requirements, few users, local usage and then later expand them to full-scale, enterprise–wide applications on large scale–out landscapes that support huge numbers of users on multiple devices.

SAP HANA application can be built to run on-premise or in the cloud, or a combination of both. They can also be built for on-premise deployment and then, without code changes needed, can be redeployed to the cloud. This also works the other way round — bring a cloud deployment on-premise.

SAP has built an active application development community who share ideas and code, enabling them to learn from each other. There are plenty of online resources to help you develop skills in native SAP HANA application development, and, of course, there is the free of charge small footprint edition of SAP HANA called **SAP HANA Express edition**, so you can get started right away even using a basic specification laptop.

When you build applications in SAP HANA, you have access to a rich application development platform which is tightly integrated with the in-memory SAP HANA database to enable high-performance applications.

From XS to XSA

There is a little history to cover here just so we avoid confusion when you see references to XS Classic (XSC) and (XSA) XS Advanced.

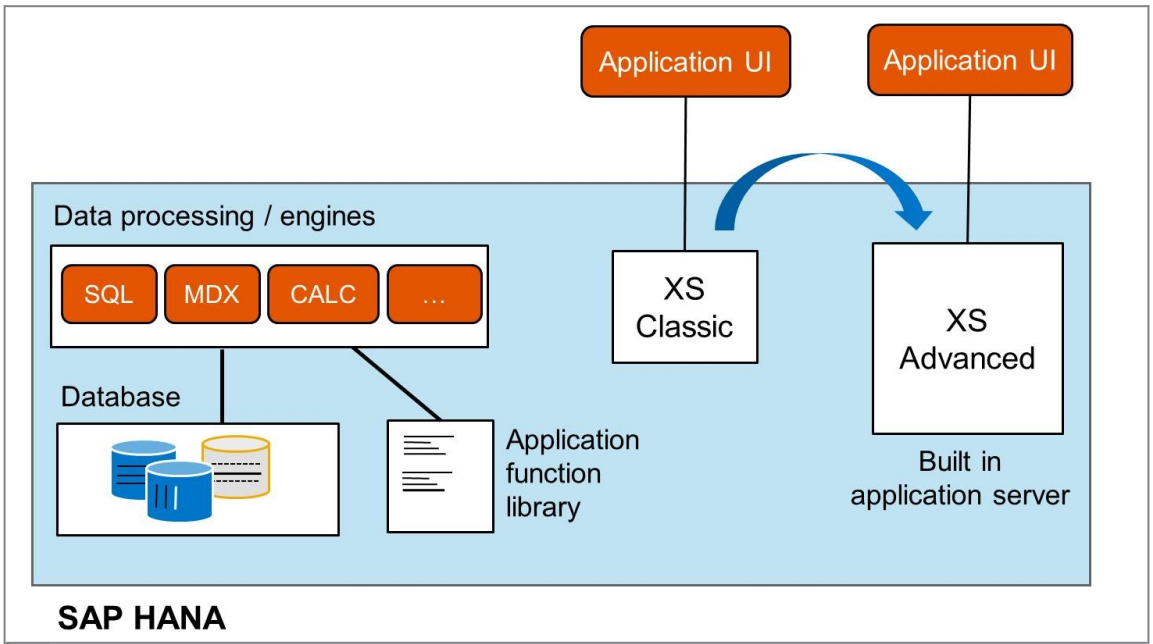


Figure 121: Evolution of the Built-In Application Server

Extended Application Services (XS) was first introduced with SAP HANA SPS05. Back then, it was 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 runtime 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). To avoid confusion, XS was then referred to as XS Classic or simply XSC. Although XSC and XSA have the same overall goal, they are technically very

different and can be considered two completely separate components. Applications developed with XSC are not compatible with XSA and vice-versa.

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

So why did SAP implement a completely new application server framework (XSA) to replace XSC?

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

Support running multiple versions of the same application — When an XSC application is built, it can only be deployed to one target run-time using fixed database schema coding. Using the new HDI approach, we can now write schema-free code and this means applications can be easily redeployed multiple times to different containers with complete isolation. The approach supports running multiple versions of the same application in the same infrastructure.

Supports more languages — Instead of just one application language used in XSC (JavaScript), XSA supports more development languages (JavaScript, Node.JS, JAVA. Plus, you can also create your own custom run-time to support more languages such as C ++, Python, Pearl, 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.

Improve security — With XSC, all users were assigned a database user with the required privileges to access database objects and data content. This approach, while nice and simple, is not acceptable to many customers who want separation between database access and application users.. With XSA, only the generated internal ‘technical’ database users are given access to the SAP HANA database. Application users’ security is handled at the application layer which in turn passes the database requests to the technical user.

In a nut shell, XSC was simply too restrictive, did not follow enough open standards, and could not support full-scale enterprise applications built for a multi-cloud/on-premise hybrid world.

Define the Data Layer

Before we can begin developing application code, we need to have the basic database objects in place that make up the data definition layer, for example: tables and views. Although it is possible to create tables and views using well-known SQL Data Definition Language (DDL) statements such as `CREATE TABLE` directly in the SQL Console of SAP HANA, this is not recommended for database objects that are part of an application. Instead you should always define your database objects using source files in the Web IDE Development View. When you build the source file, the run-time database object is generated automatically.

SAP provides two languages for the definition of the database objects:

- SQL data definition language
- Core Data Services language

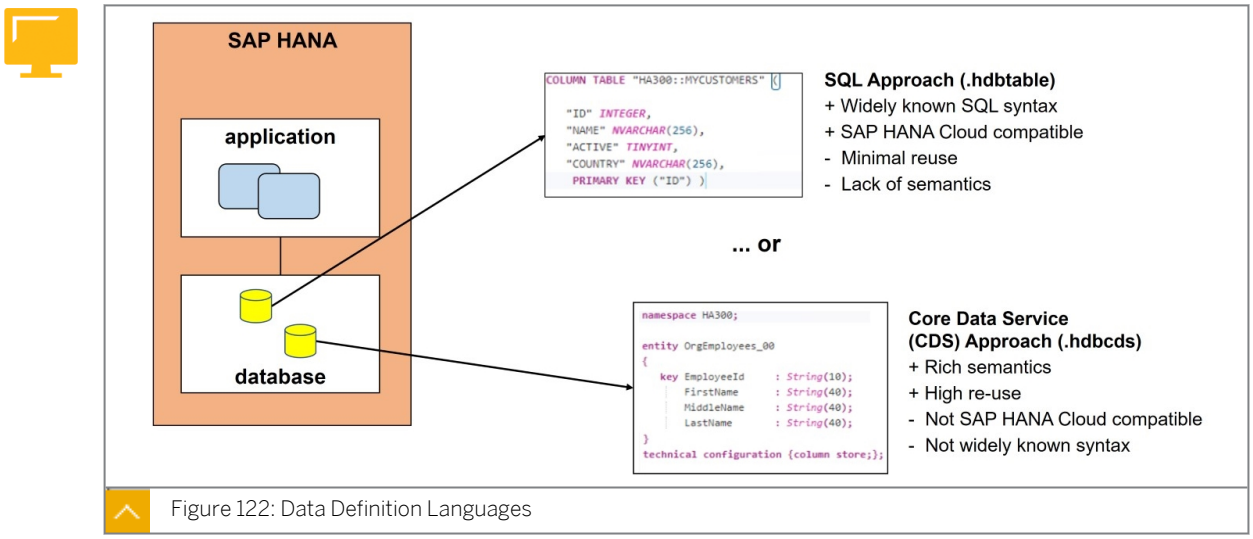


Figure 122: Data Definition Languages

You should choose to implement one of these languages and not mix them.

For the SQL approach, SAP provide multiple file types for each type of database artifact. For example, to define a table with the name SALES, your complete source file name would be: *SALES.hdbtable* and for a view, the full name would be *SALES.hdbview*. All possible database objects in SAP HANA are represented with their own source file type.

In the source file you would use the same syntax as standard SQL table and view creation. The benefit to using source files is that we have a means to capture the data definition and keep all source files of an entire application together. When we build the application, all dependencies are checked and this includes the data definition layer. It is important to remember that in XS Advanced development, a table or view must have a source file representation, otherwise the application build will fail.

Using the SQL data definition language means developers who are used to working in database development environments are likely to be familiar with the syntax and can get started quickly building tables and views.

The alternative approach to using SQL language is to define the data layer using Core Data Services (CDS).

Core Data Services is a language developed by SAP that allows the developers to define highly re-useable data types, structures, tables and views. One of the main features of CDS is that as well as the basic definition of the data, you can add rich semantics. For example, in a table definition, for each column that defines a monetary value, you could describe its currency or

even point to a column in the table that provides the currency. You could also add rules that determine how an object should be consumed. For example, a view could be defined as being consumable by another view and not by an application directly. CDS also allows multiple database objects to be creates in one source file. You could even create versions of the same table for use in different business contexts, for example, a common partner table for sales and purchasing contexts with some special features for each context.

But the downside to choosing to implement CDS is that is not a widely know language and skills in CDS development will have to be built.

Also, be aware that SAP HANA CDS is not a language that can be ported to the SAP HANA Cloud. SAP Cloud does have a similar concept to CDS but this is not SAP HANA CDS. So if your plans include moving your application to SAP HANA Cloud then you might want to consider using the standard SQL approach.

Working with SQLScript For Database Development

To ensure the best possible performance of applications, developers should push as much data processing to the SAP HANA database as possible. This can be achieved by writing as much code as possible in the database rather than in the application layer.

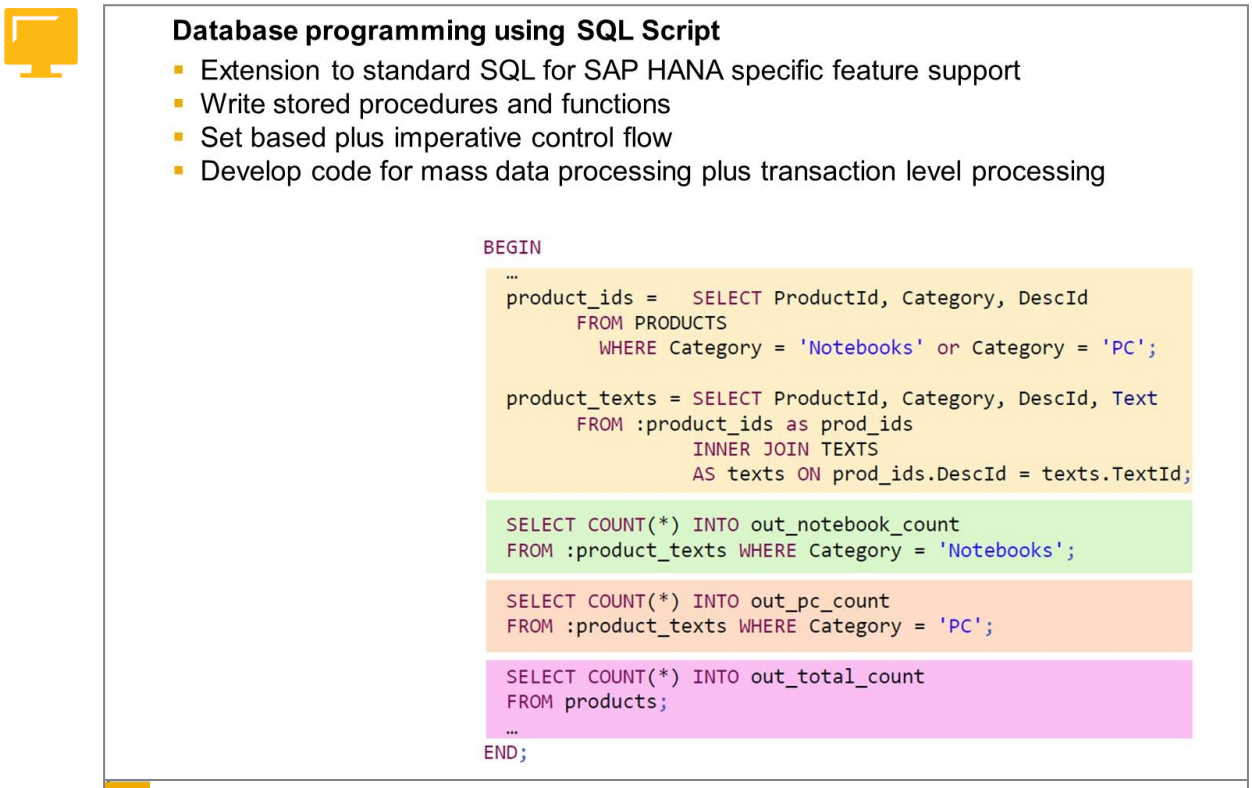


Figure 123: SQL Script for Database Development

SAP HANA provides the database language **SQLScript** to support this approach. SQLScript is a native SAP HANA language that builds on the well-known ANSI-92 SQL standard syntax and adds SAP HANA-specific features to make use of the special performance optimizations and advanced data processing capabilities of the in-memory database.

These additional features include:

- Variables to store interim results so that code can be simplified into smaller chunks. The use of variables also encourages automatic parallelization

- Conditional code such as `DO ..WHILE` and `IF - THEN - ELSE` so that program logic can be added to SQL.
- Input and output parameters that support passing of values between application code and SQL.

The SQLScript editor in Web IDE for SAP HANA provides many productivity aids including:

- Code auto-completion to speed up coding.
- Real-time error identification as you type (not just syntax errors but warnings to indicate where invalid objects are being referred to (tables columns).
- Source code debugger to step through code.
- Code Analyzer to identify weak code and provide recommendations for improvement (for example, unused variables).
- Code libraries provided by SAP and maintained by customers so that snippets of common code can be stored and reused.

The two main DDL artifacts where SQLScript is used are:

- Procedures (.hdbprocedure)
- Scalar and table functions (.hdbfunction)

The procedures and functions can be called from the application code. This supports the push down principle where all data processing should be carried out in SAP HANA database using functions and procedures and not in the application in order to achieve the best possible performance.

Tools for Developers of Native SAP HANA Applications

Web IDE for SAP HANA

The Web IDE for SAP HANA is the main tool used by developers for XS Advanced development projects.

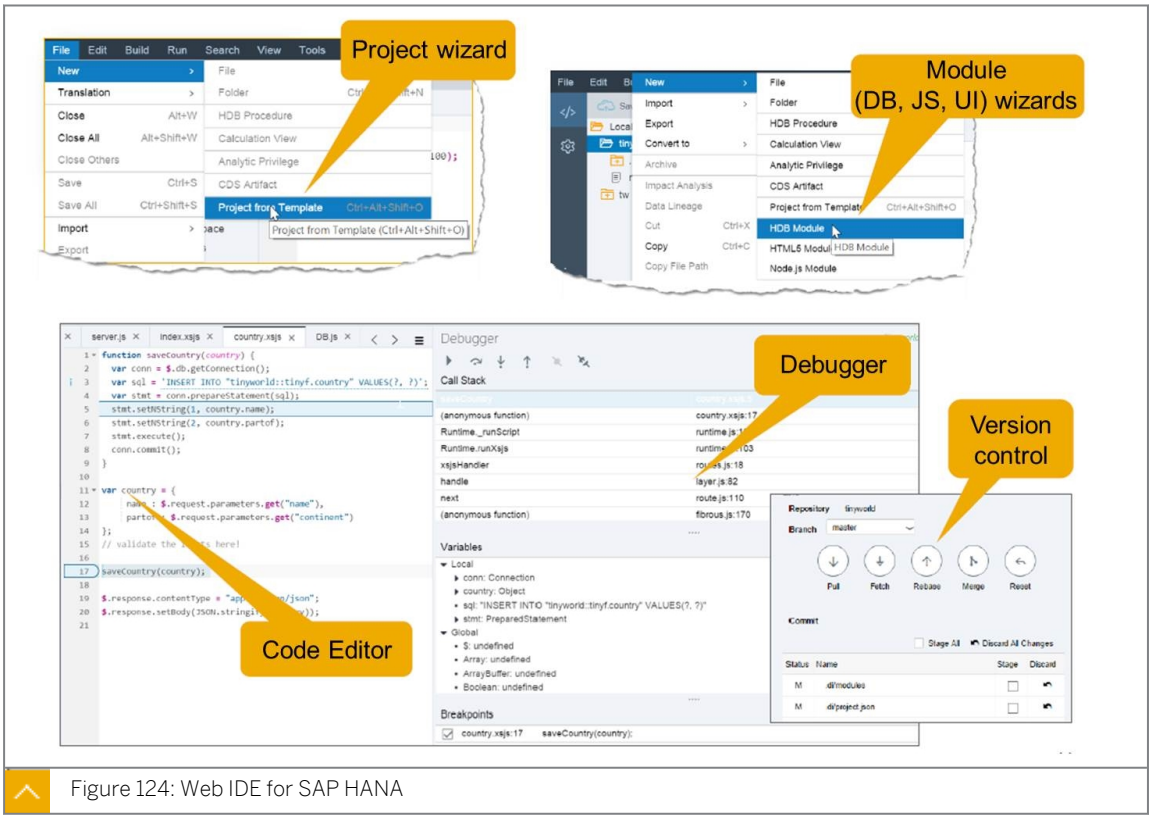


Figure 124: Web IDE for SAP HANA

The following features are provided and aimed at optimizing productivity for the developer:

- Project Wizard — get started with a new project using a built-in wizard that automates the creation of the key building blocks of an application.
- Code-specific modules — select from the provided modules to get started building code using the text and graphic editors relevant for each type of module, for example Node.JS module, HTML module.
- Code Editor — a smart editor that provides real-time syntax checking, end of line code completion, code snippet templates, and database object dependency checks in real time.
- Debugger — step through code and identify errors by viewing variable values.
- Version Control — built-in Git control panel for managing source code versions and staging and committing code.

Connect directly to the SAP HANA database with Web IDE Database Explorer

- Execute SQL directly in console
- Choose from container or catalog view

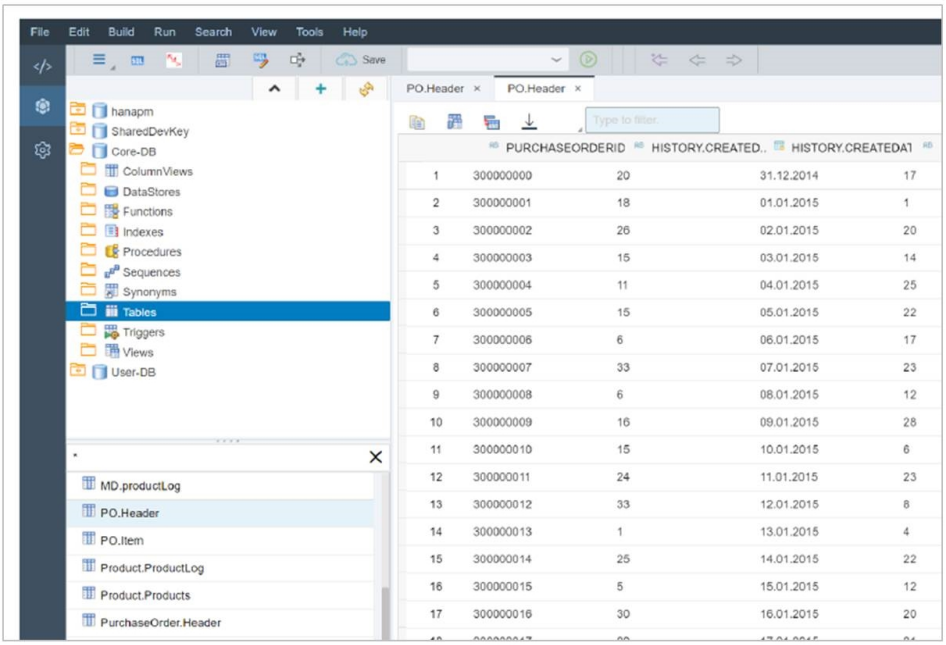


Figure 125: Connect Directly to SAP HANA Database

The Web IDE for SAP HANA includes a **Database Explorer** tool to directly connect to the SAP HANA database. This provides visibility of the database artifacts.

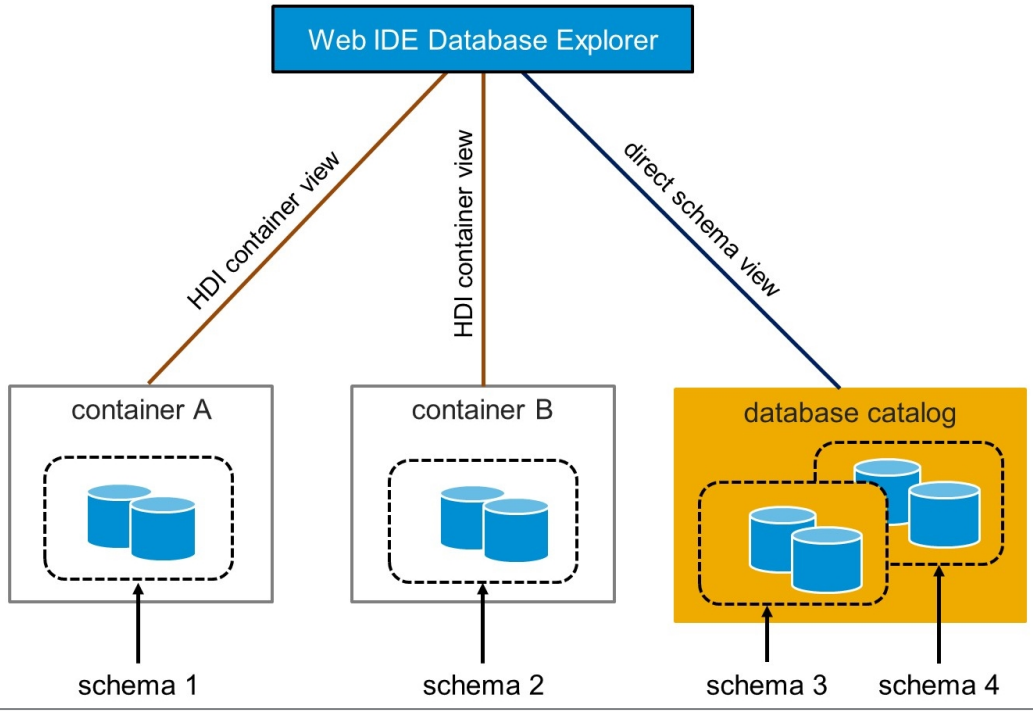


Figure 126: The HDI Container or Schema View

It is possible to connect directly to the SAP HANA database using either the classic catalog / schema approach, or by connecting to the database via the XS Advanced generated run-time

containers. Using the catalog / schema approach provides an unrestricted view of the entire database across multiple applications (subject to having the sufficient SQL privileges). The container approach is useful to view the local database objects that are built as part of a specific XS Advanced application.

Source Code Management

If you work in application development, you already appreciate the challenges of working in a team where each of the members is working on a different part of the application. Team members lock each other when they need to access common objects, or even worse their developments are overwritten by others.

Things are made more complex when the application needs to provide new features very frequently, and at the same time bugs and fixes must be dealt with urgently on the current versions of the application.

Finally, a key challenge is tracing the history of not just one piece of code but all the dependant objects at any moment in history.

With the first-generation application server (XS), source code was managed completely inside SAP HANA. A source code repository was used and developers checked in and checked out their code.

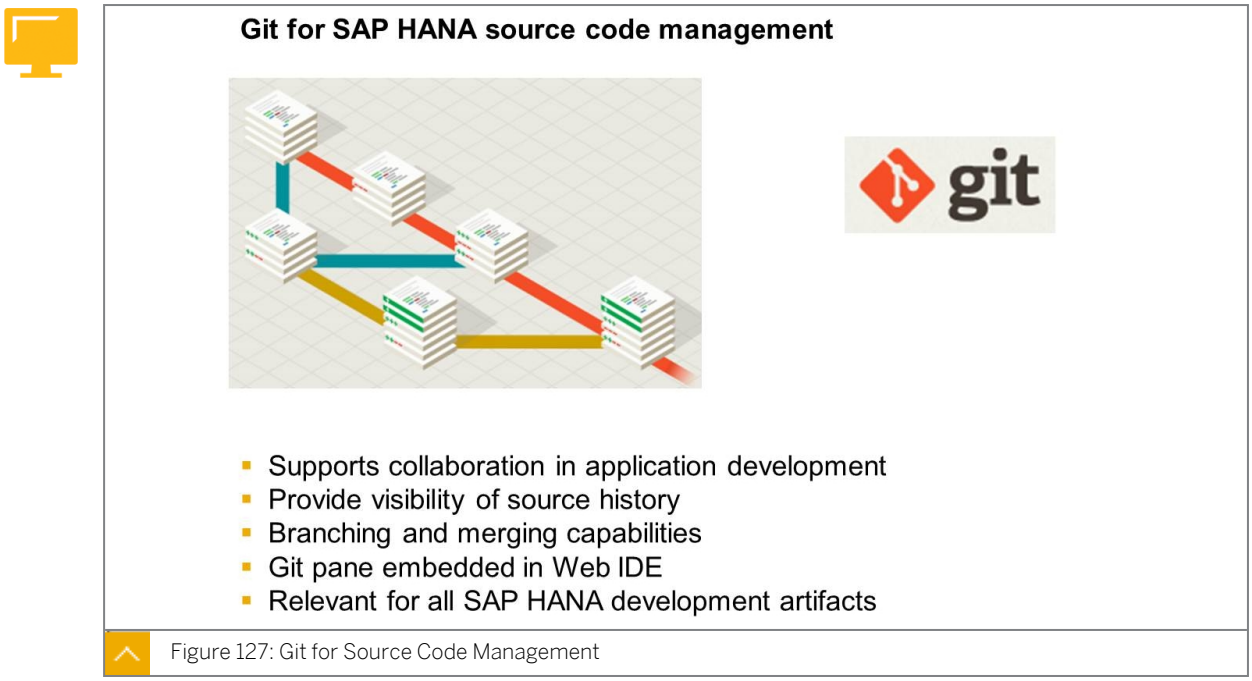


Figure 127: Git for Source Code Management

SAP HANA XSA uses Git for source code management. This includes all application and database code plus calculation views (as technically they are also stored as text files). Git has established itself as the leading source code management tool and is open source. Many application developers are already using Git and so they are able to get started right away in a familiar environment.

Git provides sophisticated branching and merging capabilities that does not rely on locking objects. Multiple developers can work on their own branches of the code in parallel and then merge the results when they are finished. Developers can pick up the version of the application at any stage of development.

There are code review tools that allow conflicts to be managed, for example, when a source file is modified by two developers and we need to figure out which version to use, or perhaps how we might manually adopt both the code from one into the other.

SAP has developed a Git console right inside the Web IDE for SAP HANA, so that all source artifacts can be seen with a Git status, such as staged, committed, and so on. It is recall old versions of source code as the history paths for all code branches is shown graphically.

In addition to Git, you can also implement Gerrit or GitHub which provides additional capabilities especially in the area of community collaboration which even extends outside your organization. A lot of publicly available code snippets are available on GitHub and Gerrit for downloading. Plus, of course, it is easy to push your code to GitHub/Gerrit for sharing.

So, with a well-established industry standard for source code management, SAP decided to give up providing these tool inside SAP HANA and use Git. SAP can now focus on adding value in other areas.

Note:
There are many online resources to learn Git, including some great videos. SAP provide some of these.

Building Applications on XS Advanced

The architecture of XS Advanced is based on the following:

- Cloud Foundry architecture
- Support for multiple development languages
- Micro-services architecture
- Git source code management
- HANA Deployment Infrastructure (HDI)

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 XS Advanced was the desire to unify the architecture of solutions built in the cloud and on-premise. Cloud Foundry provides scalability options and flexible run-times 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 on-premise and cloud using XS Classic . One of the main advantages of Cloud Foundry is that we can build applications using multiple development languages and also microservices architecture (see next). This means that we are not restricted to JavaScript, but can also work with other languages including Java, node.JS, and C++. The emphasis on Cloud Foundry is portability of applications.

Multiple Development Languages

SAP HANA XS Advanced provides full native development support for the application languages **JavaScript** (on Node.js), and **Java** (on TomEE).

Of course, XS Advanced provides run-time engines to support these languages but also many other popular run-times are supported including PHP, Pearl, Ruby, Python, C++.

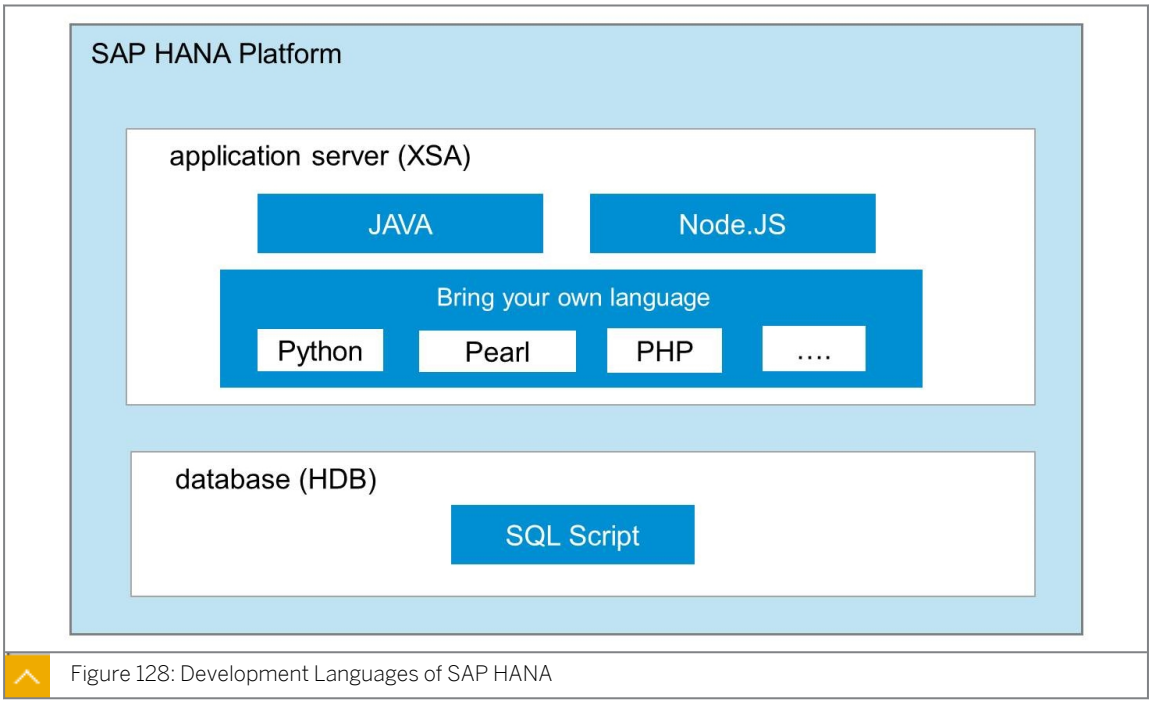


Figure 128: Development Languages of SAP HANA

To support **OData**, which is the SAP-recommended protocol for REST-based data access. XS advanced applications model the database objects that must be exposed to client requests by annotating CDS views or defining OData artifacts; the applications can then make use of these generic data providers that interpret these models.

SAP HANA now supports **Hibernate**. Hibernate is an open source object relational mapping (ORM) tool that provides a framework to map object-oriented domain models (such as sales orders or vendors) to relational databases. SAP HANA is one such relational database that is supported. Many developers now work in Hibernate in their Java projects and are familiar with this modern tool. Hibernate provides an abstraction layer between the application and the database to make it possible to address, access, and manipulate objects without having to consider how they are related to their data sources.

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 are combined using REST APIs. The modules can be developed in any language (see the figure, Development Languages of SAP HANA) 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 run-time 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 Router** orchestrates the different run-time modules that make up the complete application. Microservices have their own development and maintenance lifecycle, so you can upgrade one service independently of other services.

Git

Git is one of the industry-leading source code 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 commit, stage, fetch, merge 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 (such as tables, functions) in source code files that are organized into **modules** using Web IDE for SAP HANA.. A module looks like a top level folder and specifies the language of the source code such as Node.JS, or HTML. There is always at least one module dedicated to the database artifacts. This is the HANA Database Module (HDB).

A group of modules sit under a development **project**. To generate the run-time objects from our design time files, we **build** an application from all the source code sitting in the modules under a project. A build ensures that all object dependencies are checked and if there was even one small error in any source file, or a missing file, the entire application build fails. This is called the ‘all-or-nothing’ approach and ensures that all source code files of an application are committed together. It is possible to build isolated run-time objects by selecting an individual source file. But again, its dependencies are strictly checked and any required, but not yet available objects, cause the build to fail.

The generated run-time database objects of an application are organized in a **database container**. These containers provide isolation of the database objects that are used in different versions of the same application. We can redeploy the same application, with different features and enhancements and each application version has its own database container. 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 Classic. With XS Classic, we could only deploy one version of an application at a time as there was no 'containerization' concept.

Source Code Organization

Within each development module, we can freely create additional sub-folders in a hierarchy structure to organize our source code.

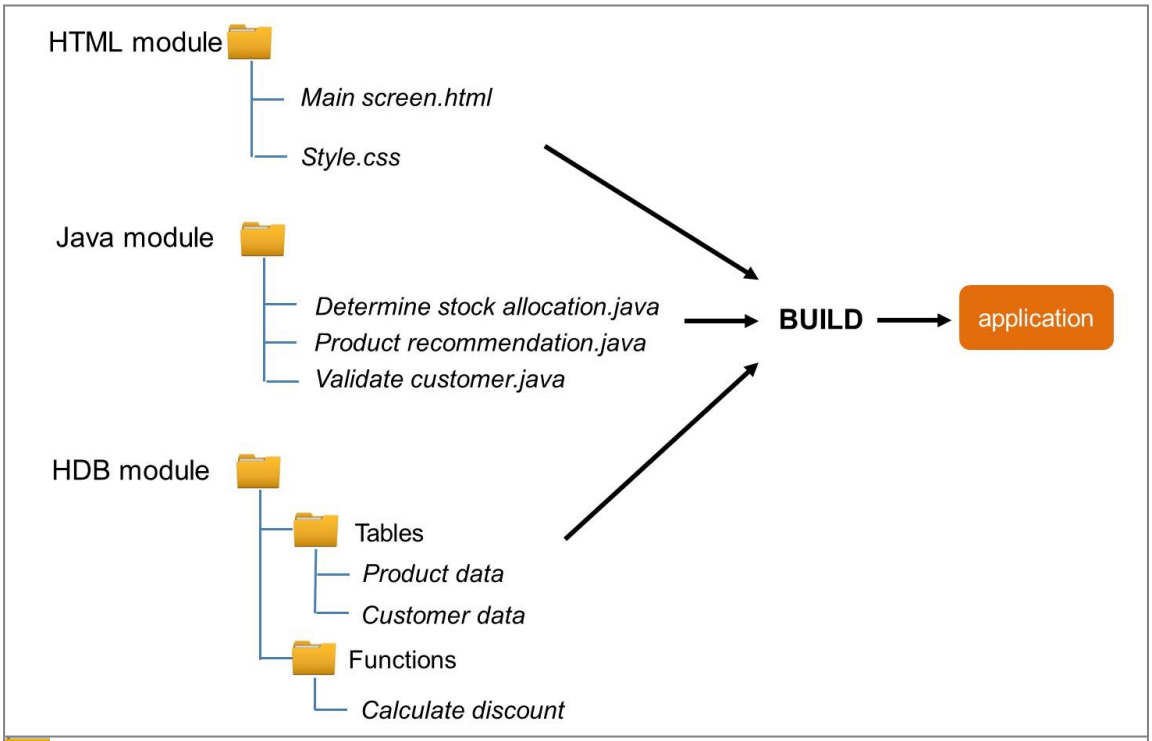


Figure 129: Source Code Organization In XS Advanced

For example, under an HTML5 module we might create a folder called *STYLES*, and in there we store the .css files. Also under the HTML5 module we might have a folder called *IMAGES* where we store the graphics used on the interface (such as buttons).

When using Git, you also see the Git status control icons alongside each source file to show whether it has been staged or committed.

It is easy to **export** any source file, or a complete folder, or even a whole project, so it can easily be **imported** to another folder in the same SAP HANA system or even an completely different SAP HANA systems. This makes the sending of source files to support teams for debugging, very easy. The source files are text-based files so they are very portable and require no special software to open them and display contained code.

Note:
To learn more about native application development on SAP HANA, you may want to consider the course HA450, *Application Development for SAP HANA*.

Application Function Libraries

You can dramatically increase performance by executing complex computations in the database instead of at the application sever level. SAP HANA provides several techniques to move application logic into the database, and one of the most important is the use of application functions delivered by SAP.

Application functions are like database procedures written in C++ and can be called with parameters to perform data intensive and complex operations. SAP delivers ready-made functions that are grouped into an **Application Function Library (AFL)**.

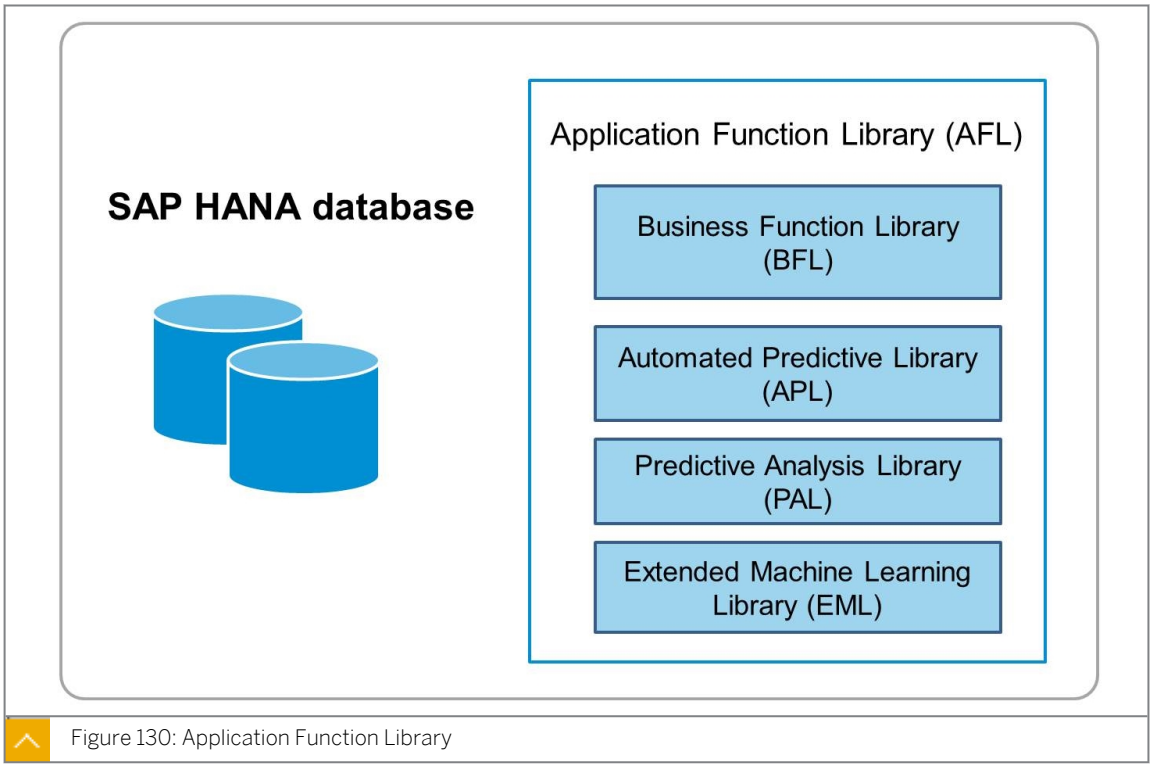


Figure 130: Application Function Library

Application Function Library (AFL) uses the following libraries:

- Business Function Library (BFL) — contains prebuilt parameter-driven functions that can be embedded in applications to perform common business functions.
- Predictive Analysis Library (PAL) — contains optimized predictive algorithms that are supplied with SPA HANA and run natively inside the SAP HANA in-memory database.
- Automated Predictive Library (APL) — provides access to the additional algorithms that are packaged with the SAP Predictive Analytics tool, but again, the algorithms run natively in SAP HANA.
- Extended Machine Learning Library (EML) — introduced with SAP HANA 2.0 SPS02. functions, provides access to external machine learning libraries. The first external library is Google TensorFlow, but more may be added.

SHINE - The SAP HANA Reference Application

SAP HANA Interactive Education (SHINE) is a fully built model application developed and maintained by SAP and is delivered free of charge and can be installed in SAP HANA. SHINE enables new SAP HANA developers to learn and develop applications running on top of the SAP HANA platform.

SAP HANA SHINE

- Prebuilt, fully working reference application based on XSA
- Learn best practices from SAP
- Learn how to work with key data processing techniques (text, spatial, and so on.)
- Constantly updated by SAP to include new features of SAP HANA

Figure 131: SAP HANA SHINE

SHINE content is designed and built upon the EPM (Enterprise Procurement Model) framework developed by SAP, and it includes all of the data models, tables, views, dashboards and so on, with a real enterprise use case. It also comes with embedded documentation that describes the building blocks of the applicator from a technical perspective as well as operating instructions. There are also plenty of online resources to guide you with the installation and exploration of SHINE.

SHINE include analytics, text, spatial, and even transactional (CRUD) applications. The idea is that you learn from these fully functioning and documented applications instead of struggling to get something working from an empty system.

SHINE even comes with its own data generator so you can generate and load data into the tables used by SHINE.

Note:
There is a version of SHINE for XS as well as a version for XS Advanced, so be sure to download the correct version.

Developers should download SAP HANA, express edition and install SHINE so that they have a quick start environment to build their native SAP HANA application development skills. There are tutorials on-line to guide the install of SAP HANA Express and SHINE.

Migrating XS Classic Applications to XS Advanced

Since SAP 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.

However, customers have already built many applications in XS, the original application development and runtime 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

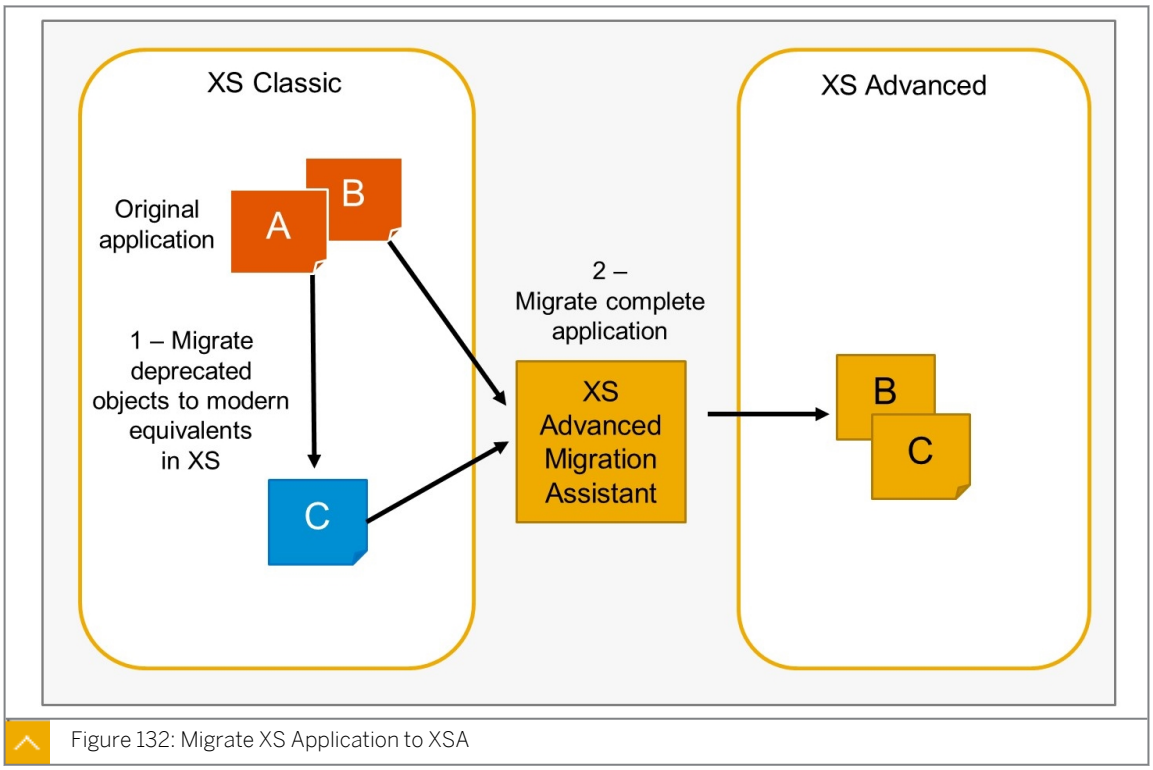


Figure 132: Migrate XS Application to XSA

To support the migration process, SAP provides 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 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 delivers 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 has developed this SHINE capability in order to illustrate and explain the typical challenges you are likely to encounter when migrating XS classic applications.

LESSON SUMMARY

You should now be able to:

- Describe an SAP HANA Native Application

Learning Assessment

1. What are the tasks of the ABAP developer in an SAP HANA implementation?

   Choose the correct answers.

   - ☐ **A** Adjust migrated ABAP code so it functions correctly on SAP HANA.
   - ☐ **B** Look for opportunities to use ABAP code that works on any database.
   - ☐ **C** Speed up existing ABAP applications using SAP HANA optimizations.
   - ☐ **D** Develop new ABAP applications that exploit SAP HANA advanced capabilities.

2. For ABAP applications running on SAP HANA, the recommended approach is to bring data to code.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

3. Which tool is used for XS Advanced application development?

   Choose the correct answer.

   - ☐ **A** SAP HANA Web-based Workbench
   - ☐ **B** SAP Web IDE for SAP HANA
   - ☐ **C** SAP HANA Studio

4. Applications developed with XS classic must be migrated to XSA before they can run in SAP HANA 2.0.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

5. What does XS Advanced application development use for source code management?

   Choose the correct answer.

   - [ ] A HDI
   - [ ] B Node.JS
   - [ ] C Git
   - [ ] D Cloud Foundry

Learning Assessment - Answers

1. What are the tasks of the ABAP developer in an SAP HANA implementation?

   Choose the correct answers.

   - [X] A Adjust migrated ABAP code so it functions correctly on SAP HANA.
   - [ ] B Look for opportunities to use ABAP code that works on any database.
   - [X] C Speed up existing ABAP applications using SAP HANA optimizations.
   - [X] D Develop new ABAP applications that exploit SAP HANA advanced capabilities.

   Correct! Amongst other things, ABAP developers are responsible for adjusting migrated ABAP code so it functions correctly on SAP HANA, speeding up existing ABAP applications using SAP HANA optimizations and developing new ABAP applications that exploit SAP HANA advanced capabilities. ABAP developers do not look for opportunities to use ABAP code that works on any database. Code should always be written to use only native SAP HANA database features. There will be no other database that needs to be considered when writing ABAP code.

2. For ABAP applications running on SAP HANA, the recommended approach is to bring data to code.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   Correct! In fact it is the other way round: Code to data.

3. Which tool is used for XS Advanced application development?

   Choose the correct answer.

   - [ ] A SAP HANA Web-based Workbench
   - [X] B SAP Web IDE for SAP HANA
   - [ ] C SAP HANA Studio

   Correct — SAP Web IDE for SAP HANA is used for XS Advanced application development. SAP HANA Studio and Web-based Workbench are used only XS classic application development only.

4. Applications developed with XS classic must be migrated to XSA before they can run in SAP HANA 2.0.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct — SAP HANA 2.0 supports XS classic applications without adjustment; they do not need to be migrated to XS Advanced.

5. What does XS Advanced application development use for source code management?

   Choose the correct answer.

   ☐ A HDI

   ☐ B Node.JS

   ☒ C Git

   ☐ D Cloud Foundry

   Correct — The answer is Git. HDI is used to manage the deployment of run-time objects. Node.JS is a programming language. Cloud Foundry is the open source framework adopted by XS Advanced and is used to provide architecture consistency across cloud developments.

UNIT 8 Monitoring SAP HANA

UNIT OBJECTIVES

- SAP HANA Cockpit
- Monitoring SAP HANA

Unit 8
Lesson 1

Introducing SAP HANA Cockpit

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

- SAP HANA Cockpit

SAP HANA Cockpit

The SAP HANA Cockpit provides a single point of access to aggregate, system, and database administration features. Examples of these include database monitoring, user management, and data backup.

SAP HANA 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.

SAP HANA Cockpit is a web-based HTML5 user interface that you access through a browser. It runs on SAP HANA XSA.

Caution:
Don't use SAP HANA Studio for administration of SAP HANA database systems as this tool is deprecated and will be removed in the near future.

Introducing SAP HANA Cockpit 2.0

The SAP HANA cockpit provides tools for the administration and monitoring of SAP HANA databases, and for limited development capabilities through the SAP HANA database explorer. You can manage multiple databases, each running version SAP HANA 1.0 SPS 12, or later.

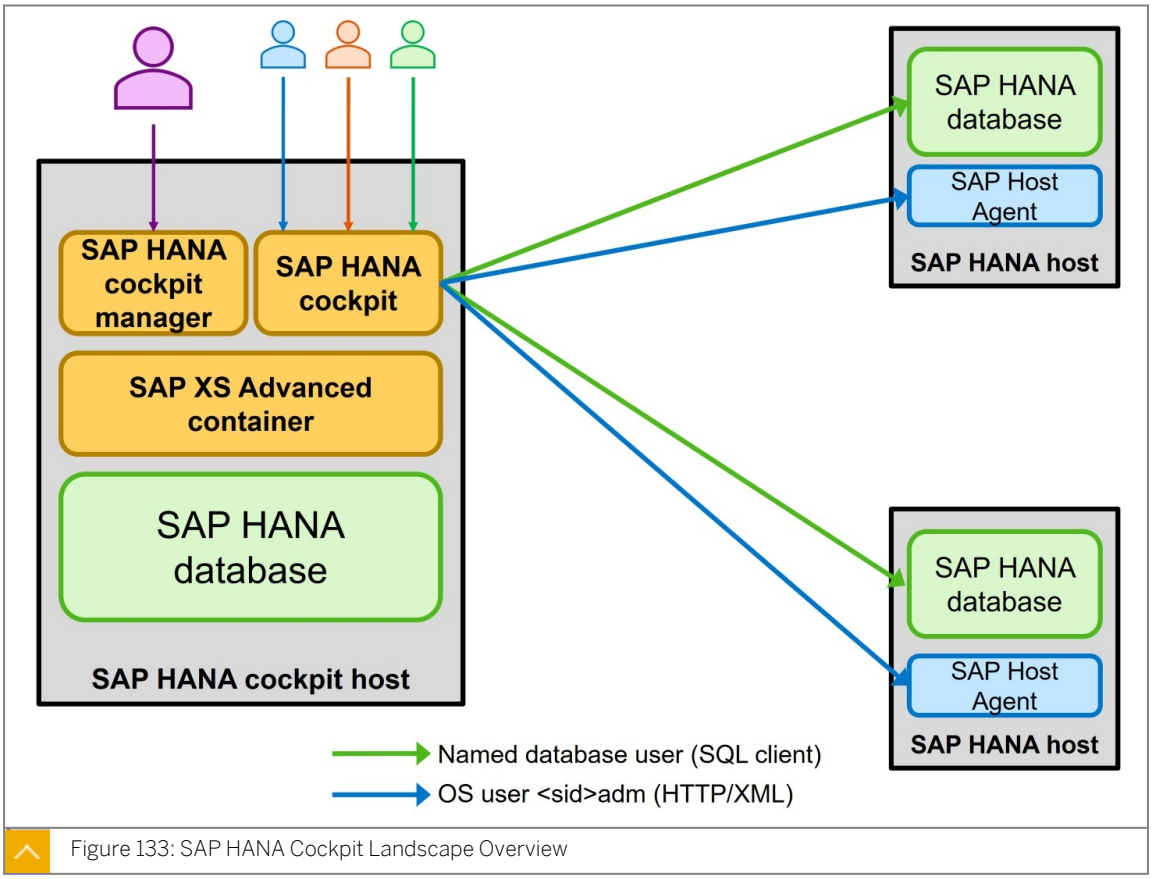


Figure 133: SAP HANA Cockpit Landscape Overview

Note:
SAP HANA cockpit runs on an SAP HANA express database, which is included in the installation. The SAP HANA cockpit can also be installed in an existing SAP HANA system in a separate tenant database (shared database).

SAP HANA cockpit consists of two parts:

- The SAP HANA cockpit manager
- The SAP HANA cockpit

SAP HANA Cockpit Manager

To get started with the SAP HANA Cockpit, you need to create administration key users, register the databases that need to be monitored and assign administrator and databases to groups. When a SAP HANA administrator logs on to the SAP HANA cockpit, he will see only the systems that he is assigned to. This configuration is done using a separate tool called the *SAP HANA Cockpit Manager*.

With the SAP HANA cockpit manager, you can register databases and create groups of databases that other cockpit users can access within the SAP HANA cockpit. The databases are managed through the cockpit administrator (cockpit_admin) user. This user is created during the installation of SAP HANA cockpit and is assigned the master password provided during the installation.

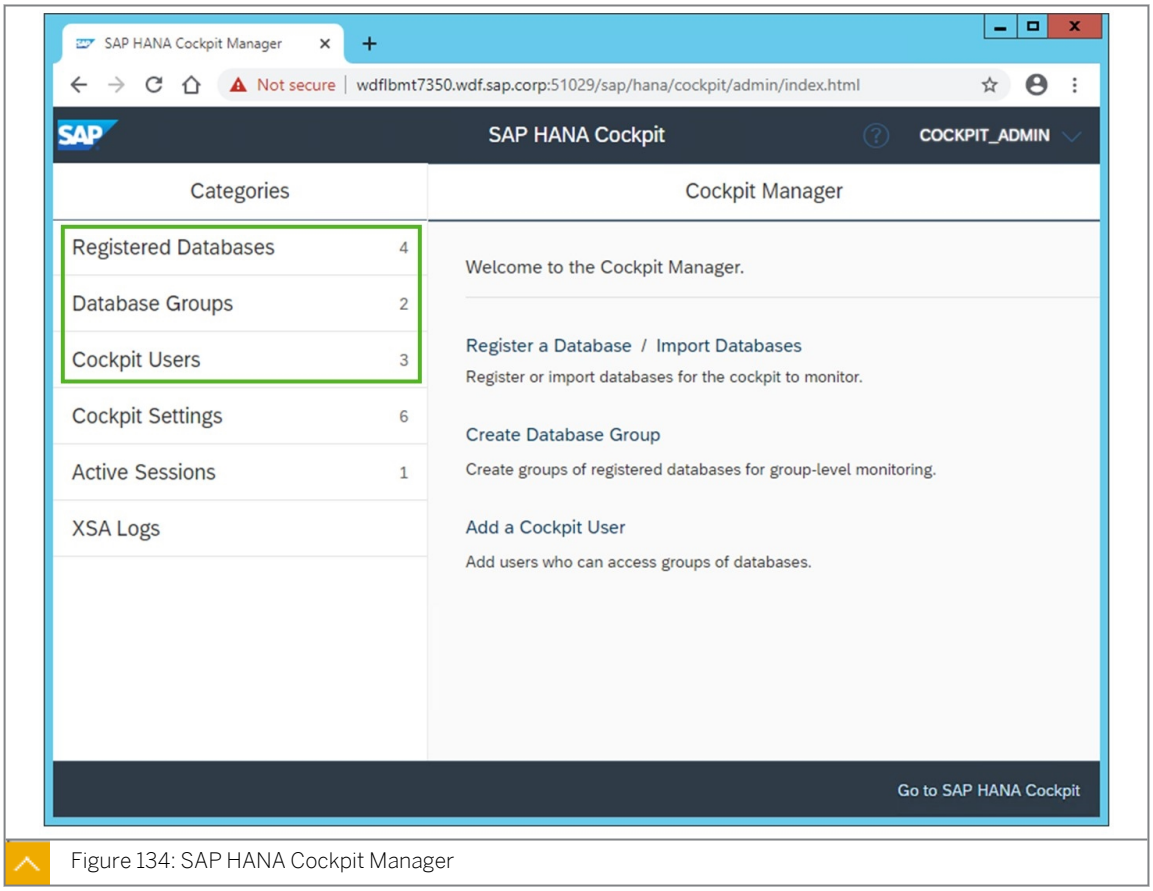


Figure 134: SAP HANA Cockpit Manager

A database is the system database or a tenant in the SAP HANA system, identified by a host, instance number and tenant name. Suppose that a business unit has set up a new SAP HANA system and wants it to be managed through the cockpit. The first step is to register the databases, that is, the system database and the tenant(s) from that SAP HANA system, in the cockpit.

SAP HANA Cockpit

The SAP HANA cockpit provides tools for the administration and monitoring of SAP HANA databases (databases), and for development capabilities through the SAP HANA database explorer. You can manage multiple databases, each running version SAP HANA 1.0 SPS 12, or later. databases running version SAP HANA 2.0 SPS 01 or later run in multi-container mode, but you can also monitor single-container systems running earlier versions of SAP HANA.

The SAP HANA cockpit provides aggregate, system and database administration features, such as database monitoring, user management, and data backup. You can use the SAP HANA cockpit to start and stop systems or services, monitor the system, configure system settings, and manage users and authorizations.

Cockpit apps 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 page of an individual resource. Links, data, tiles, and different parts of a single tile will provide more detailed information and functions when selected.

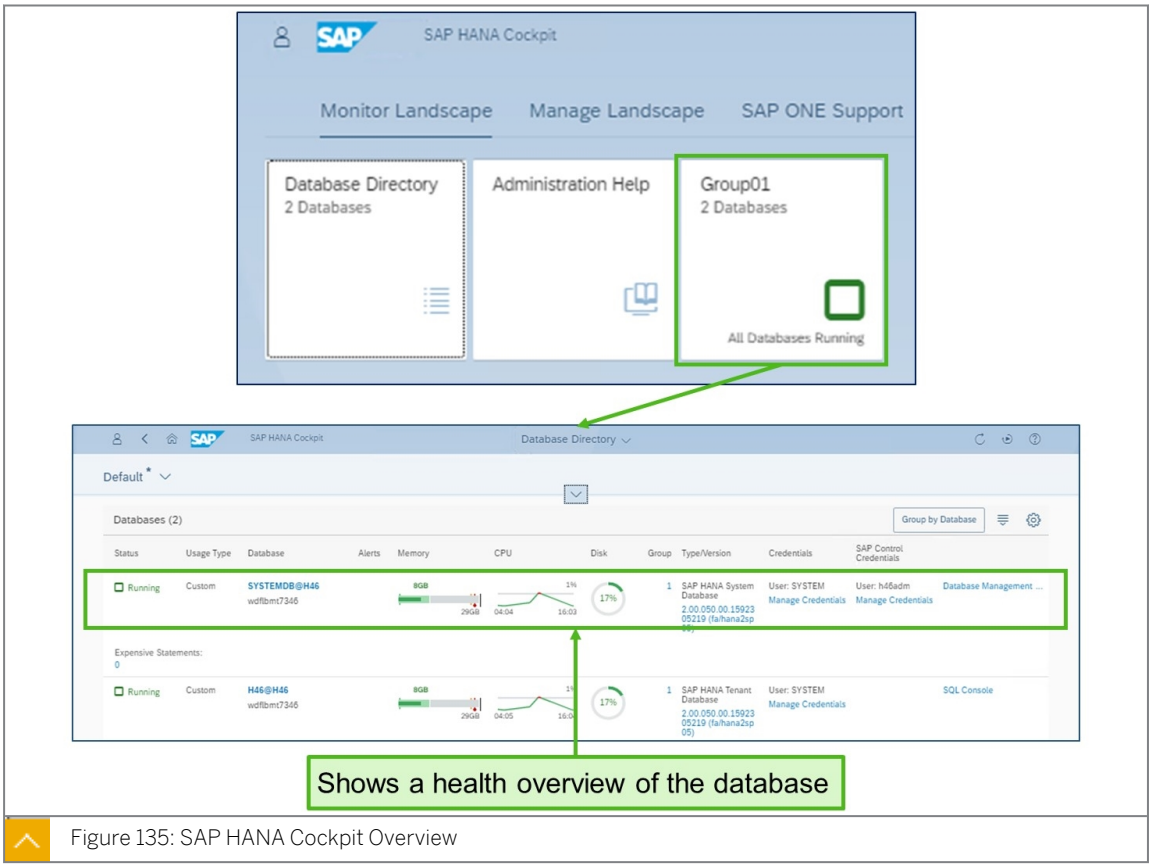


Figure 135: SAP HANA Cockpit Overview

When you first launch the cockpit, you can see system and tenant databases. (The cockpit refers to these as databases). A database is an SAP HANA system (identified by a host name and instance number) which may be a system or tenant database in a tenant (database) container, or a system in a single database container. These databases are organized into database groups - you'll only see databases belonging to the groups to which your cockpit user has been granted access. At a glance, you can see top alerts from more than one database, compare database configurations and monitor the health of multiple databases.

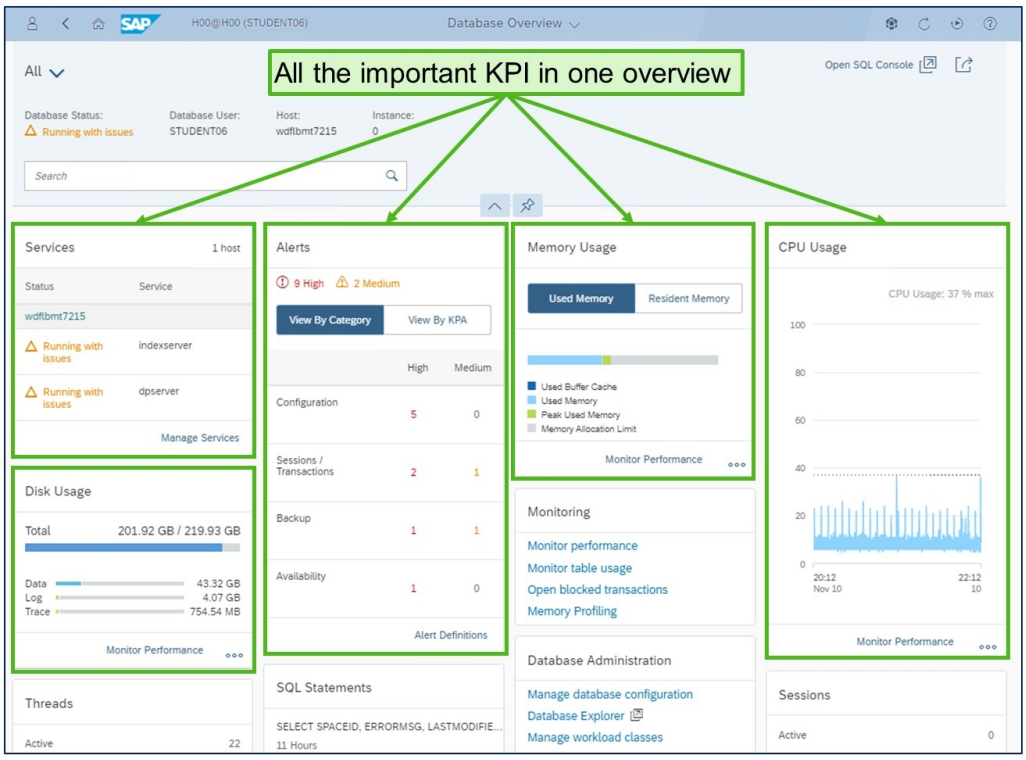


Figure 136: SAP HANA Cockpit Overview

Whenever you like, you can drill down to perform in-depth monitoring on an individual system or tenant. In order to see alerts and other data for this individual database you need to enter database user credentials. These database user credentials must pre-exist (that is, they're already created on the database you're drilling into), and must have the system privilege CATALOG READ and SELECT on _SYS_STATISTICS. For any systems running version SAP HANA 2.0 SPS 01, or later, the cockpit database administrator has the option to enable or enforce single sign-on (SSO).

SAP HANA Database Explorer

Integrated into the cockpit is the SAP HANA database explorer. The database explorer provides the ability to query information about the database using SQL statements, as well as the ability view information about your database's catalog objects.

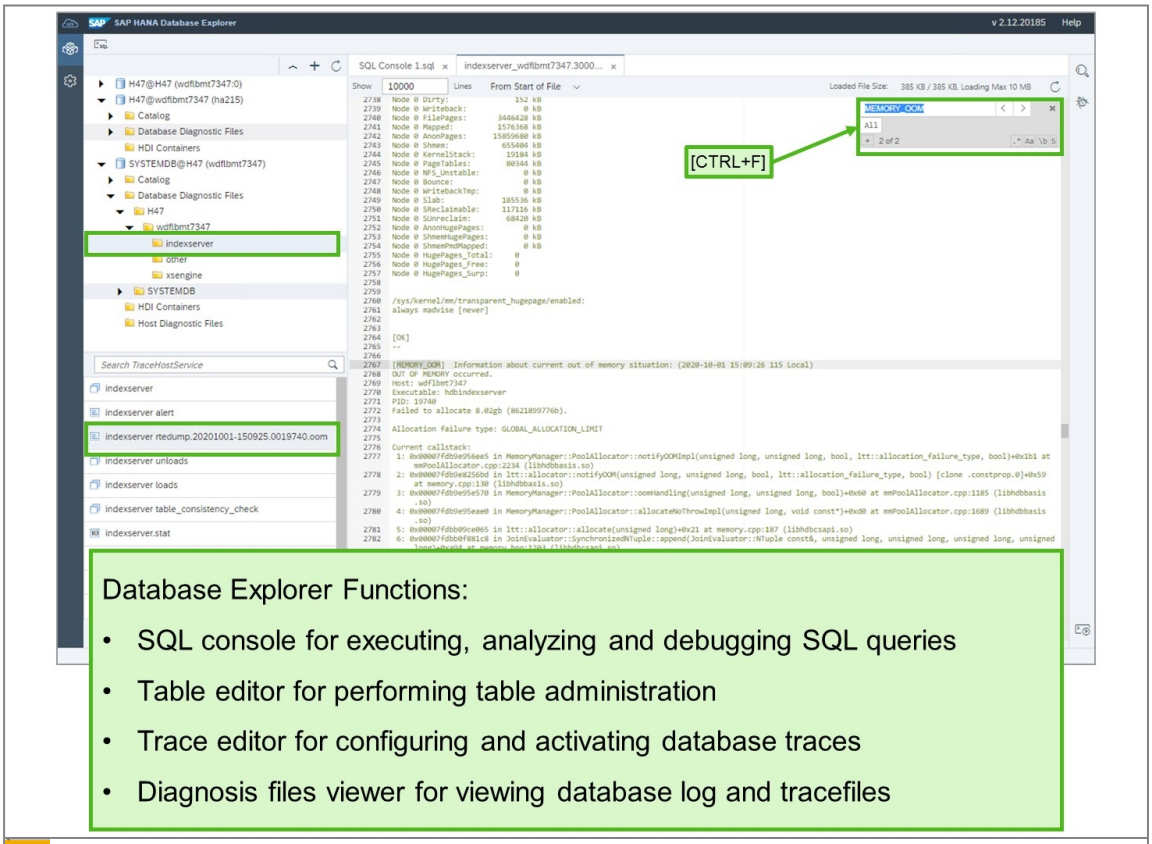


Figure 137: SAP HANA Database Explorer

What Can I do with the SAP HANA Database Explorer?

The SAP HANA database explorer allows you to query your database and view information about your catalog objects, by providing the following functionality:

- An SQL console

  Create SQLScript procedures and queries, and then execute them or analyze their performance using the SQL analyzer.

- An SQL analyzer

  View detailed information on your queries and evaluate potential bottlenecks and optimizations for these queries. The SQL analyzer is accessible from the SQL console, as well as from the plan trace and expensive statement features in the SAP HANA cockpit.

- An SQL debugger

  View the call stack, set break points, view and evaluate expressions and variables.

- Trace Configuration

  Database tracing is always active. This means that information about error situations is always recorded. However, for more detailed analysis of a specific problem or component, you may need to configure and activate additional traces like SQL trace, expensive statements trace, plan trace, performance trace, or kernel trace.

- Table editor

The table editor provides detailed table meta data like general table definition information, columns definitions, detailed memory and disk usage, partitions information and table runtime Information.

- Diagnosis files viewer

  Diagnosis files include log and trace files, as well as a mixture of other diagnosis, error, and information files. In the event of problems with the SAP HANA database, you can check these diagnosis files for errors. Traces generally need to be explicitly enabled and configured.

LESSON SUMMARY

You should now be able to:

- SAP HANA Cockpit

Unit 8
Lesson 2

Monitoring SAP HANA

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

- Monitoring SAP HANA

SAP HANA Services

The Database Overview page contains a series of cards. Each card provides a starting point to a group of related applications that allow you to monitor and manage the database.

The cards are grouped into four views:

- Monitoring
- Security and User Management
- Administration
- All

To change the view, at the top left of the page, click the downward pointing chevron icon beside the current view name. Cockpit remembers the last view selected and automatically loads it the next time you open SAP HANA Cockpit. You can rearrange cards in a view by dragging a card to a new location, but you can't add or remove cards to or from the view.

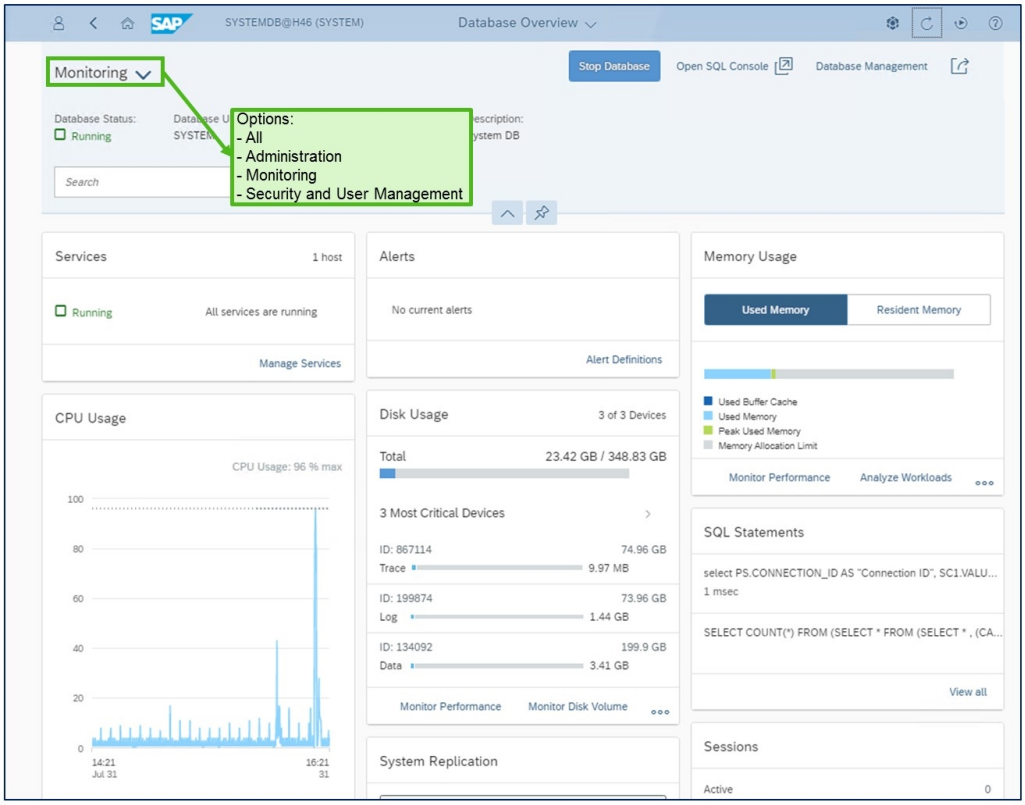


Figure 138: Database Overview Page

When moving your mouse over a card, areas that change color and displays a hand icon to indicate an available link. For example, on the Memory Usage card, a hand and a color change appear when you move over the card title, Used Memory, Resident Memory, and Monitor Performance or within the card. On the Monitoring card, nothing happens when you hover over the title, but the hand and color change happens when you hoover over each item within the card.

Some cards provide summary information about the database. This information is collected from the card's underlying applications. Depending where you click on these cards, you can change the summary information displayed, open the card's application to display filtered details, or open a related application from another card.

Manage Services application

Open the Manage Services application to view the health of the SAP HANA services. On the Manage Services page you get per service detailed information on Memory and CPU consumption, status, alerts, PID, start time and host assignment.

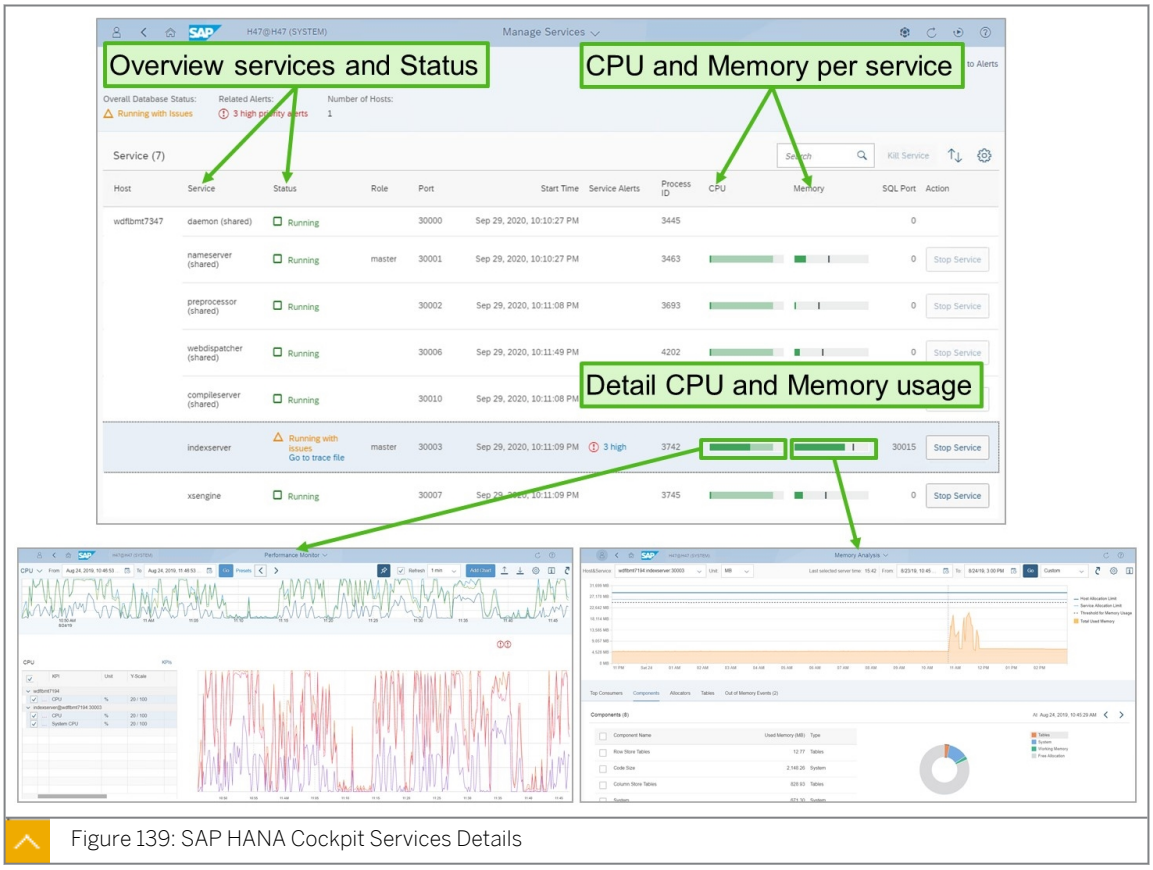


Figure 139: SAP HANA Cockpit Services Details

The available services depends on the installed SAP HANA components and features. The services can be groups in Core Services, Optional Services and SAP HANA XS Advanced Services.

Core Services

The core services are available in every installation scenario.

- Name server

  The name server, which runs in the system database only, owns the information about the topology of the SAP HANA system, including knowledge of the tenant databases that exist in the system.

- Index server

  The index server, which runs in every tenant database (but not the system database), contains the actual data stores and the engines for processing the data.

- Compile server

  The compile server performs the compilation of stored procedures and programs, for example, SQLScript procedures. It runs on every host and does not persist data. It runs in the system database and serves all tenant databases.

- Preprocessor server

  The preprocessor server is used by the index server to analyze text data and extract the information on which the text search capabilities are based. It runs in the system database and serves all tenant databases.

- SAP Web Dispatcher

The Web Dispatcher processes inbound HTTP and HTTPS connections to XS classic services.

- SAP start service

  The SAP start service is responsible for starting and stopping the other services in the correct order. It also performs other functions, such as monitoring their runtime state.

Optional Services

In addition to the core services mentioned above, depending on the installed SAP HANA components and features the following optional servers may also be running.

- Script server

  The script server is used to execute application function libraries written in C++.

- Document store server

  This server is required for the document store repository. The document store allows native operations on JSON documents and joins with other column or row store tables.

- XS advanced runtime

  SAP HANA includes a run-time environment for application development: SAP HANA extended application services, advanced model (XS advanced). The SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of previous SAP HANA extended application services, classic model (XS classic).

  The SAP HANA XS advanced runtime consists of several processes for platform services and for executing applications.

  The SAP HANA XS advanced runtime runs either on dedicated hosts or together with other SAP HANA components on the same host.

- SAP HANA Deployment Infrastructure (HDI) server

  HDI handles the deployment of design-time artifacts into the SAP HANA database. If XS advanced is installed in the system, HDI is already enabled. Otherwise, you must enable it manually.

- Extended store server

  The extended store server is part of SAP HANA dynamic tiering. It provides a high-performance disk-based column store for very big data up to the petabyte range.

- Data provisioning server

  The data provisioning server is part of SAP HANA smart data integration. It provides capabilities such as data provisioning in real time and batch mode, real-time data transformations, data quality functions, adapters for various types of remote sources, and an adapter SDK for developing additional adapters.

- Streaming cluster

  The streaming cluster is part of SAP HANA streaming analytics. Streaming analytics extends SAP HANA with capabilities of SAP Event Stream Processor for consuming data streams and complex event processing.

- Accelerator for SAP ASE

The SAP ASE server is part of SAP HANA accelerator for SAP ASE. It provides SAP Adaptive Server Enterprise (ASE) users the ability to use SAP HANA on SAP ASE data for real-time analytics.

- SAP HANA remote data sync

  The remote data sync server is part of SAP HANA real-time replication. SAP HANA remote data sync is a session-based synchronization technology designed to synchronize SAP SQL Anywhere remote databases with a consolidated database.

- XS classic server

  SAP HANA Extended Application Services, classic model (XS, classic) is the application server for native SAP HANA-based web applications. It is installed with the SAP HANA database and allows developers to write and run SAP HANA-based applications without the need to run an additional application server. SAP HANA XS is also used to run web-based tools that come with SAP HANA, for instance for administration, lifecycle management and development.

  XS classic is the original implementation of SAP HANA XS.

  The XS classic server can run as a separate server process or embedded within the index server.

Caution:
SAP HANA XS, classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

SAP HANA XS Advanced Services

SAP HANA includes a run-time environment for application development: SAP HANA extended application services, advanced model (XS advanced). The SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of previous SAP HANA extended application services, classic model (XS classic).

The SAP HANA XS advanced runtime runs either on dedicated hosts or together with other SAP HANA components on the same host. If the runtime platform of SAP HANA XS advanced is installed in your system, the following additional services run in the system database for platform services and for executing applications:

- SAP HANA XS Controller

  The Controller is the central management component of SAP HANA XS advanced. For example, it has a view on all deployed and/or running applications, and persists configuration and status information in the database.

  The Platform Router instance is managed by the xscontroller service. The Platform Router, which is realized by an SAP Web Dispatcher instance, exposes the public endpoint for the entire system.

- SAP HANA XS Execution Agent

  The Execution Agent is responsible for managing processes, that is starting, keeping alive, and stopping tasks.

- SAP HANA XS User Authentication and Authorization (UAA)

The UAA service manages user logon and logoff requests in SAP HANA XS advanced.

Monitoring SAP HANA Performance

Analyzing the memory allocation of the SAP HANA database can help you understand and resolve unusual memory usage and out-of-memory incidents.

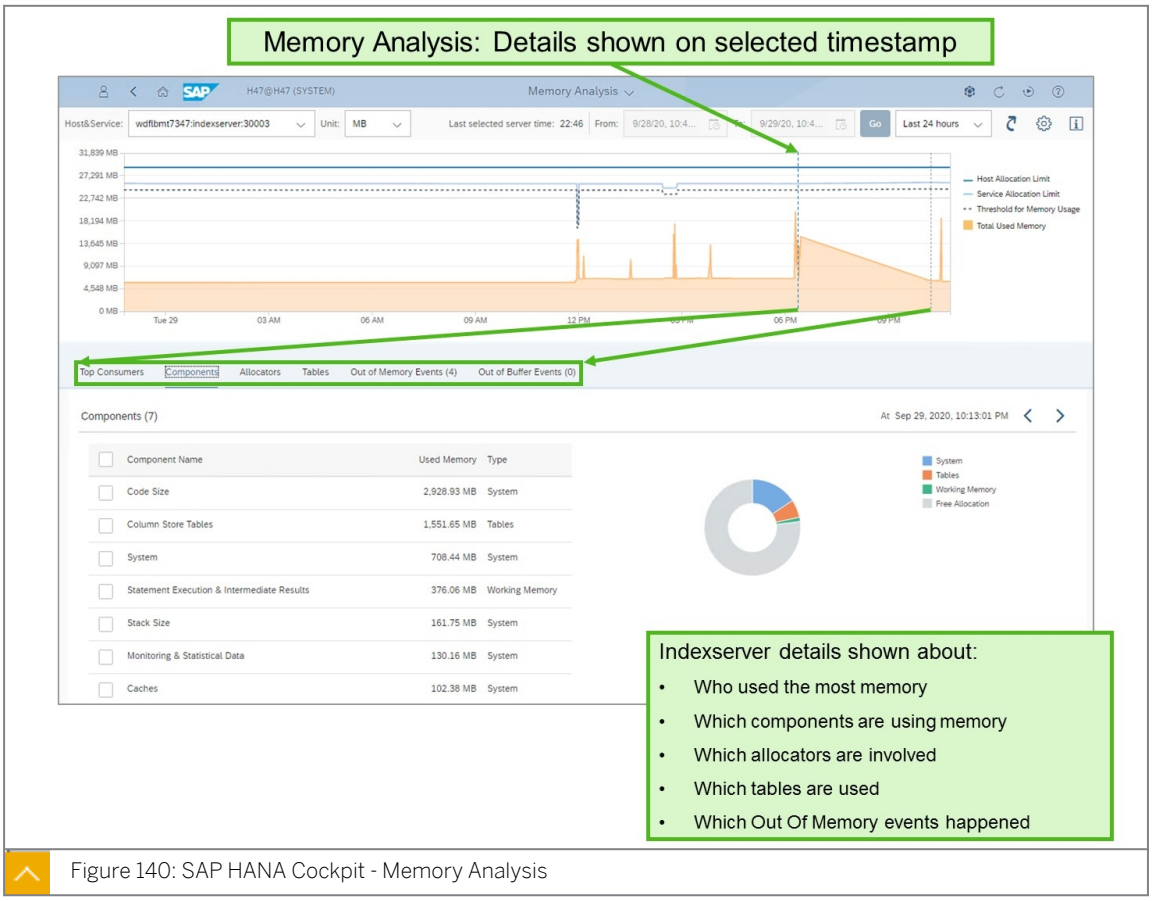


Figure 140: SAP HANA Cockpit - Memory Analysis

The *Memory Analysis* app enables you to visualize and explore the memory allocation of every service of a selected host during a specified time range. If you notice an increase in overall memory usage, you can investigate whether it's due to a particular component, allocator, or table.

The Performance Monitor

Analyzing the performance of the SAP HANA database over time can help you pinpoint bottlenecks, identify patterns, and forecast requirements. Use the Performance Monitor to visually analyze historical performance data across a range of key performance indicators related to memory, disk, and CPU usage.

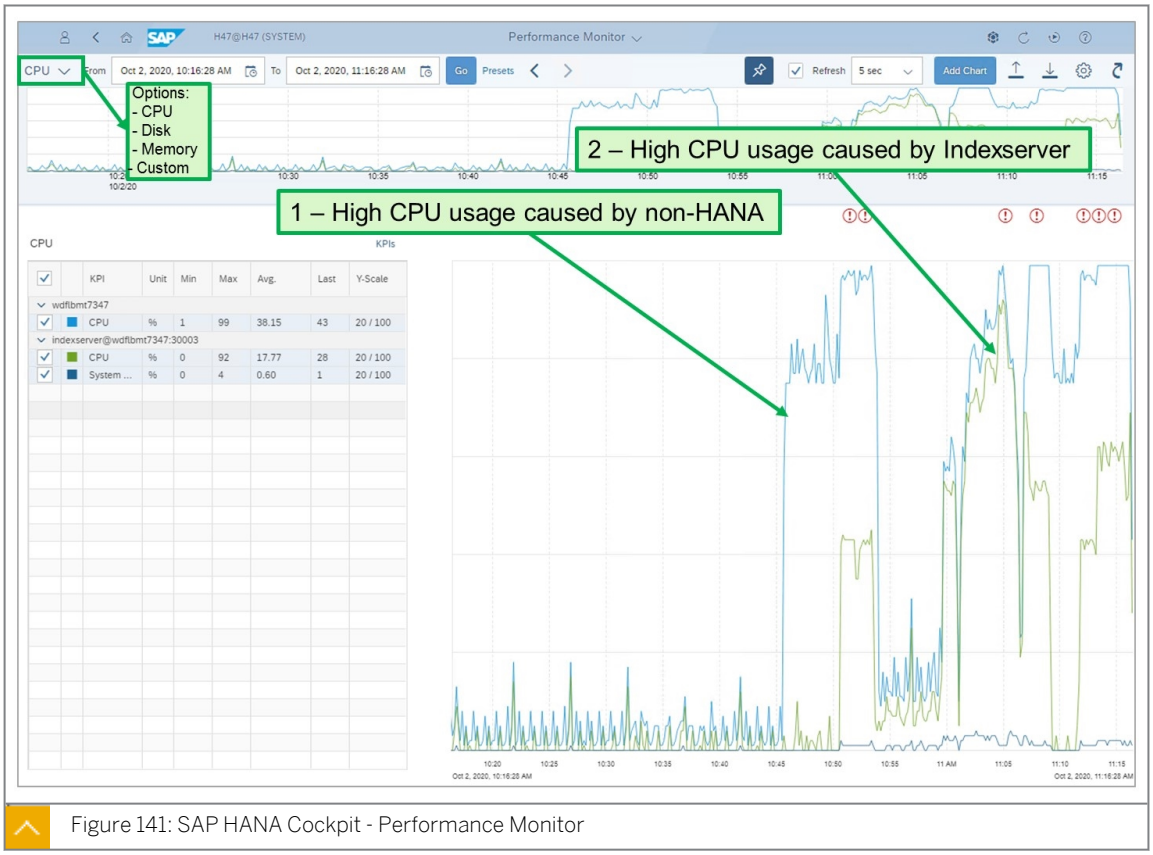


Figure 141: SAP HANA Cockpit - Performance Monitor

The Performance Monitor opens displaying the load graph for the selected resource: CPU, disk, or memory. The load graph initially visualizes resource usage of all hosts and services listed on the left according to the default KPI group of the selected resource.

Change the information displayed on the load graph by:

- Switching between a number of predefined views like Memory, CPU and Disk. You can also create and store your own views.
- Defining the monitored time frame by entering your desired dates or selecting from several presets.
- Adding charts to create custom charts displaying the host and services selection, and selected KPIs.
- Zooming into a specific time on a graph by brushing across the desired selection on the load graph directly.
- Comparing the performance of your selected KPIs at different times using the Performance Comparison page.
- Filtering the results you see in the chart by selecting only the KPIs you are interested in.

Checking SQL Statements

Use the SQL Statements application for analyzing the current most critical statements running in the SAP HANA database can help you identify the root cause of poor performance, CPU bottlenecks, or out-of-memory situations. Enabling memory tracking (Overview view only) allows you to monitor the amount of memory used by single statement executions.

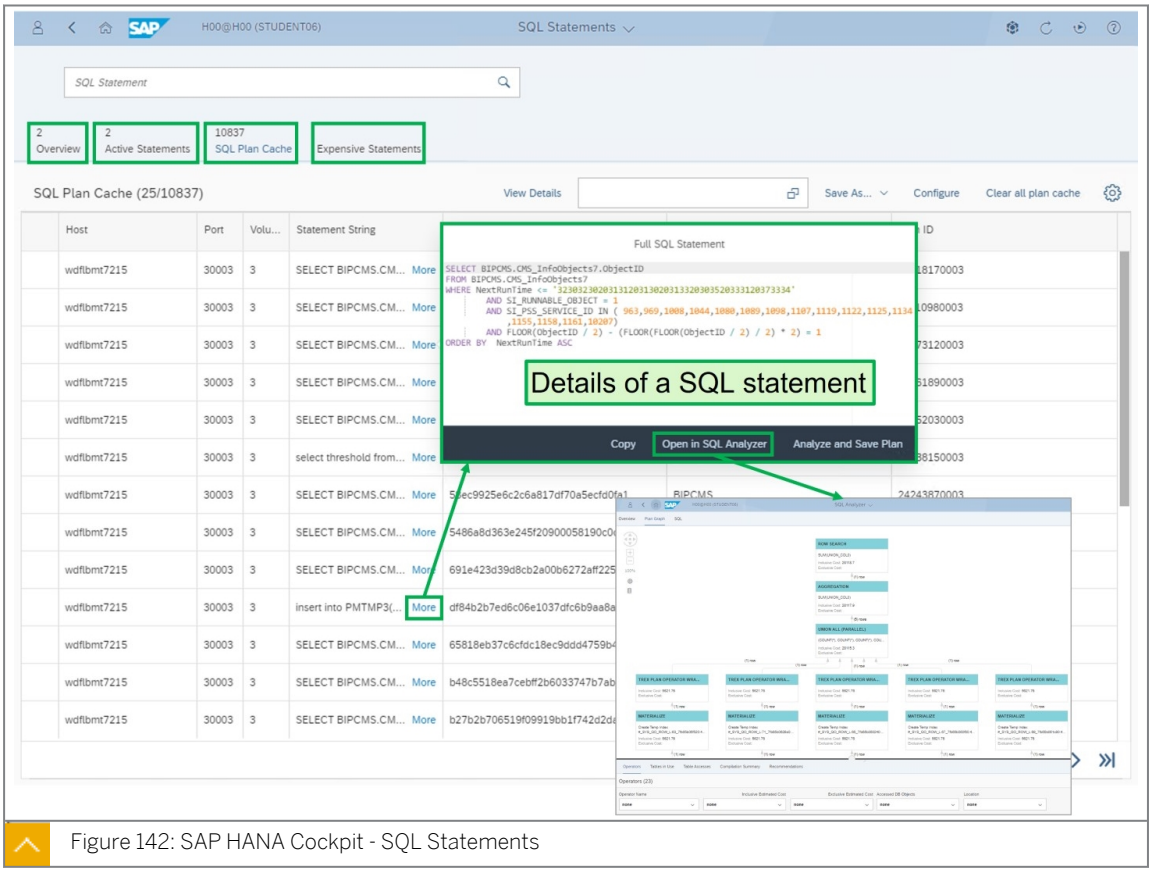


Figure 142: SAP HANA Cockpit - SQL Statements

Statements are ranked based on a combination of the following criteria:

- Runtime of the current statement execution
- Lock wait time of the current statement execution
- Cursor duration of the current statement execution

With this short overview of SAP HANA Cockpit we only touched the surface of the capabilities of the SAP HANA cockpit application. In the SAP HANA System Administration curriculum we will take a deep-dive in the administration features of SAP HANA Cockpit.

The SAP HANA System Administration curriculum consists of the following courses:

- HA200 - SAP HANA 2.0 SPS05 - Installation and Administration

  This course covers in detail all key capabilities of SAP HANA database system administration. Through the lectures and exercises you will learn how to install, update and operate an SAP HANA 2.0 SPS05 database using SAP HANA Cockpit. The course covers the most important database administration tasks of an SAP HANA system administrator. During the course, you will learn details about starting and stopping, changing the configuration parameters, monitoring the database, backup and recovery and troubleshoot of a multi-tenant SAP HANA 2.0 SPS05 database system.

- HA201 - SAP HANA 2.0 SPS05 - High Availability and Disaster Tolerance Administration

  This course covers all key capabilities of SAP HANA 2.0 SPS05 database system in the areas of High Availability and Disaster Tolerance. Through the lectures and exercises you will learn about scale-out and auto host- failover, SAP HANA storage and system replication with Active/Active. You will install and configure these setups in our lab environment.

- HA215 - SAP HANA 2.0 SPS05 - Using Monitoring and Performance Tools

  In this course you will learn about emergency trouble shooting the SAP HANA database using the SAP HANA cockpit. You will also learn how to use the SAP HANA traces to investigate for expensive SQL statements that might cause SAP HANA database performance problems in the areas of memory, CPU and disk I/O. You will also learn how to setup SAP HANA Workload Management and SAP HANA Capture and Replay tools to perform proactive monitoring and performance safeguarding.

LESSON SUMMARY

You should now be able to:

- Monitoring SAP HANA

Learning Assessment

1. What are valid user interfaces for SAP HANA Cockpit?

   Choose the correct answer.

   - ☐ **A** SSH terminal
   - ☐ **B** Web browser
   - ☐ **C** Eclipse IDE

2. Which of the following services are SAP HANA core services?

   Choose the correct answers.

   - ☐ **A** Name server
   - ☐ **B** XS advanced runtime
   - ☐ **C** Extended store server
   - ☐ **D** Preprocessor server

Learning Assessment - Answers

1. What are valid user interfaces for SAP HANA Cockpit?

   Choose the correct answer.

   - [ ] A SSH terminal
   - [x] B Web browser
   - [ ] C Eclipse IDE

   Correct. SAP HANA Cockpit is a browser based application.

2. Which of the following services are SAP HANA core services?

   Choose the correct answers.

   - [x] A Name server
   - [ ] B XS advanced runtime
   - [ ] C Extended store server
   - [x] D Preprocessor server

   Correct. The SAP HANA core services are: Name server, index server, compile server, preprocessor server, SAP web dispatcher and SAP start service. All the other services are dependant on the installed components or options.

UNIT 9 Security and Data Privacy with SAP HANA

UNIT OBJECTIVES

- Describe the security of data models
- Describe how SAP HANA Ensures Data Privacy
- Understanding Administration Security

Securing Analytics

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe the security of data models

Securing Data Models

Data models in SAP HANA are created using calculation views. Calculations views provide multiple security features to ensure the user is not able to view unauthorized data.

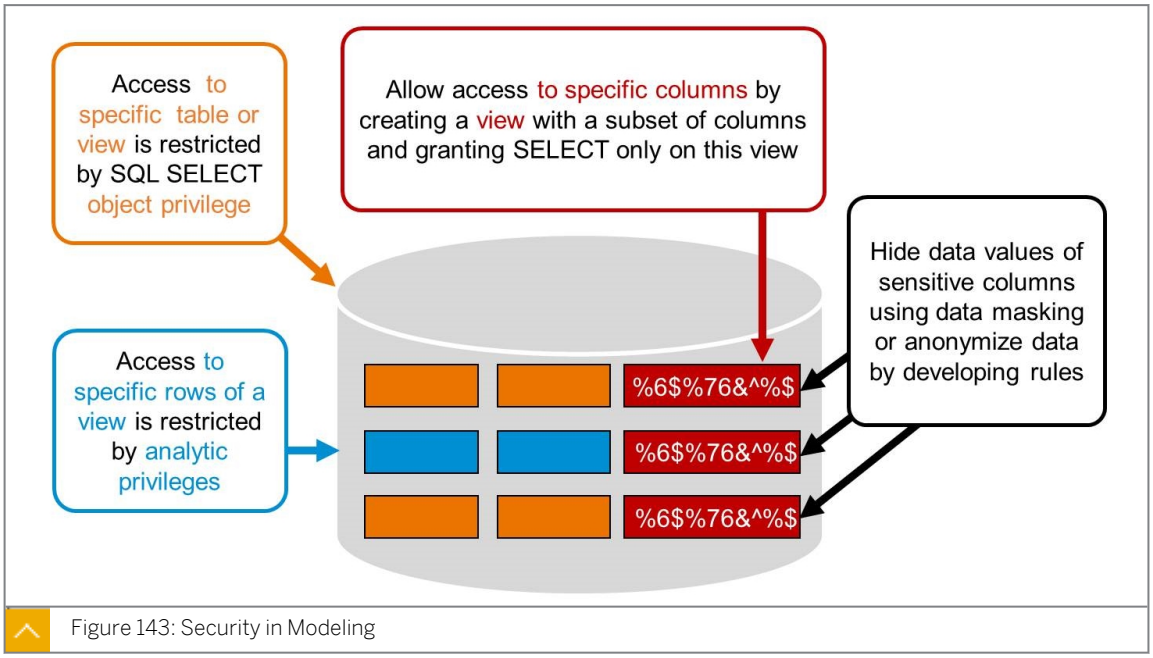


Figure 143: Security in Modeling

There are three levels of access to consider:

1. The first level of access is to the actual database objects used in the data model. This usually means source tables, procedures and functions that are consumed by the calculation view. This is achieved through standard SQL database permissions. In our case we usually only need the *select* permission on the database objects. Permissions can be granted directly to the individual user, but it is more likely that roles will be first created to organize the many users and the permissions will be granted to the roles so that they are inherited by the assigned users. This approach is more efficient for maintenance.
2. So now we can access the actual database objects used in the data model but we don't see any data! You now need to define data access at the row 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 to describe, for each calculation

view, the rules to allow data access. The conditions can be very simple such as *Barry can view any data but only for company Acme* . The conditions can also be more complex such as *Olive can view data for company Acme but only between 2017–2018 and also that relates to product group A or B*. A user/role can be assigned to multiple analytic privileges. Analytic privilege logic can be written using simple form based rules or for more complex rules we can write using SQLScript.

3. So now we can see all data within our allowed privileges, but perhaps some users should not see the entire value of a field. For sensitive data we can hide all or partial data values by applying a masking rule to each column. For example, instead of displaying a telephone number we only show *XX657 XXX764*. We can decide which users or roles should have the masking rule applied.

LESSON SUMMARY

You should now be able to:

- Describe the security of data models

Unit 9
Lesson 2

Data Privacy

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

- Describe how SAP HANA Ensures Data Privacy

Data Anonymization

Compliancy with data privacy laws is a big challenge for many organizations who would like to develop analytics over their data that includes personal information. Organizations are aware that new rules protect the individuals' identity and how their data is used. This means that often organizations are reluctant to start new projects for fear that they might be at risk of breaking data privacy laws.

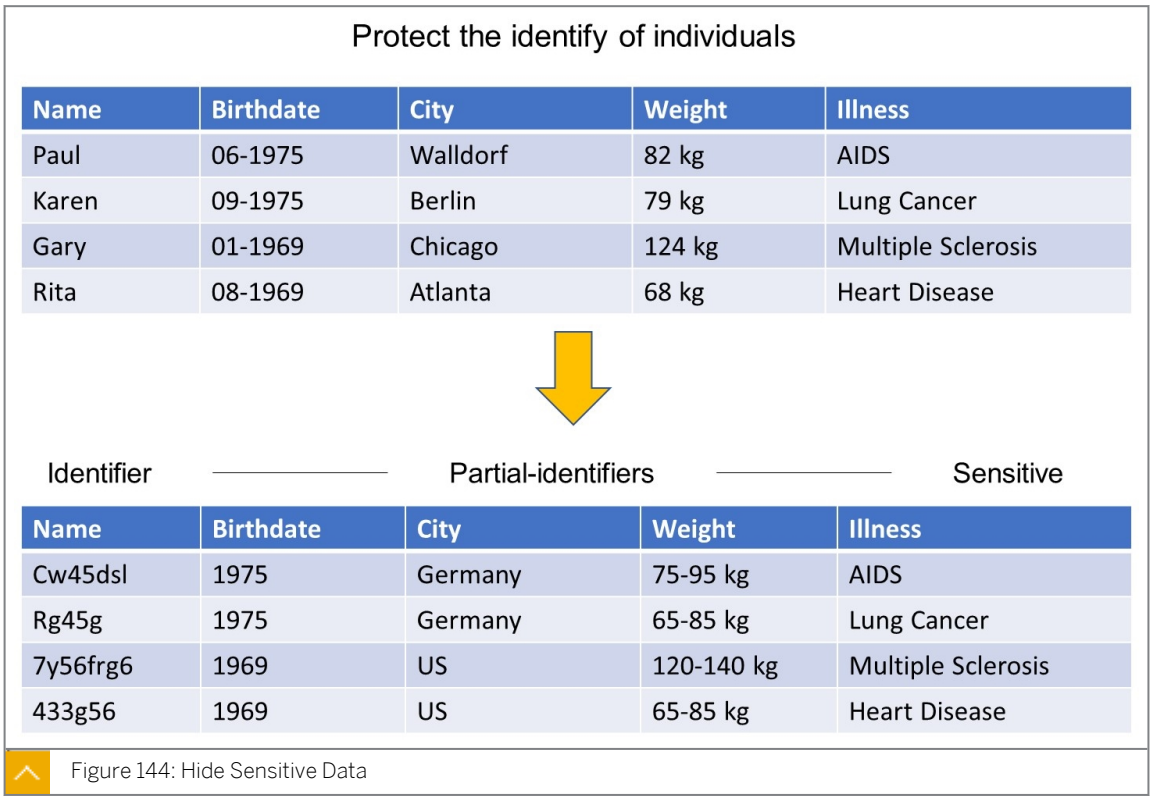
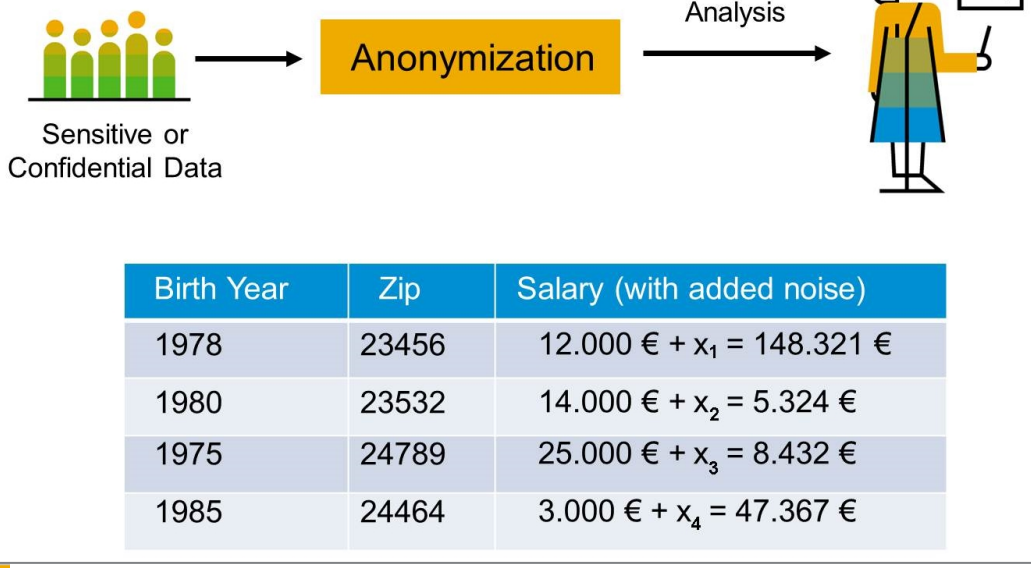


Figure 144: Hide Sensitive Data

With SAP HANA we can build sophisticated data anonymization rules into our data models so that sensitive data is not exposed. We can apply rules to sensitive data so that it is not possible to identify key entities, such as individuals, from that data. For example, if I wanted to report on total sickness days that were taken, displayed by the persons' age, if we knew the age of a particular person, and we knew there was only one person in the team of that age, it might be easy to identify that person if their age was displayed in the report. So in this case it would be better to define an anonymization rule that groups the records by a higher level age

interval, such as decade. This anonymization technique basically hides the entity in a larger group.



Birth Year	Zip	Salary (with added noise)
1978	23456	12.000 € + x_1 = 148.321 €
1980	23532	14.000 € + x_2 = 5.324 €
1975	24789	25.000 € + x_3 = 8.432 €
1985	24464	3.000 € + x_4 = 47.367 €

Figure 145: Data Anonymization

Another anonymization method is to add noise to data to make it impossible to derive any meaningful information at an individual personal level. But when aggregated, the data makes sense. For example add or subtract values to an individual's salary, but the overall additions and subtractions net to zero so what remains are the true salary values.

LESSON SUMMARY

You should now be able to:

- Describe how SAP HANA Ensures Data Privacy

Unit 9
Lesson 3

SAP HANA System Security

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

- Understanding Administration Security

SAP HANA Technical User Administration

User Lifecycle Management

Every user who wants to work directly with the SAP HANA database must have a database user with the necessary privileges. Depending on the scenario, the user accessing SAP HANA may either be a technical system user or an individual end user.

After successful logon, the user's authorization to perform the requested operations on the requested objects is verified. This is determined by the privileges that the user has been granted. Privileges can be granted to database users either directly, or indirectly through roles. Several tools are available for provisioning and managing users. For more information about the authorization model of the SAP HANA database, see the section on authorization.

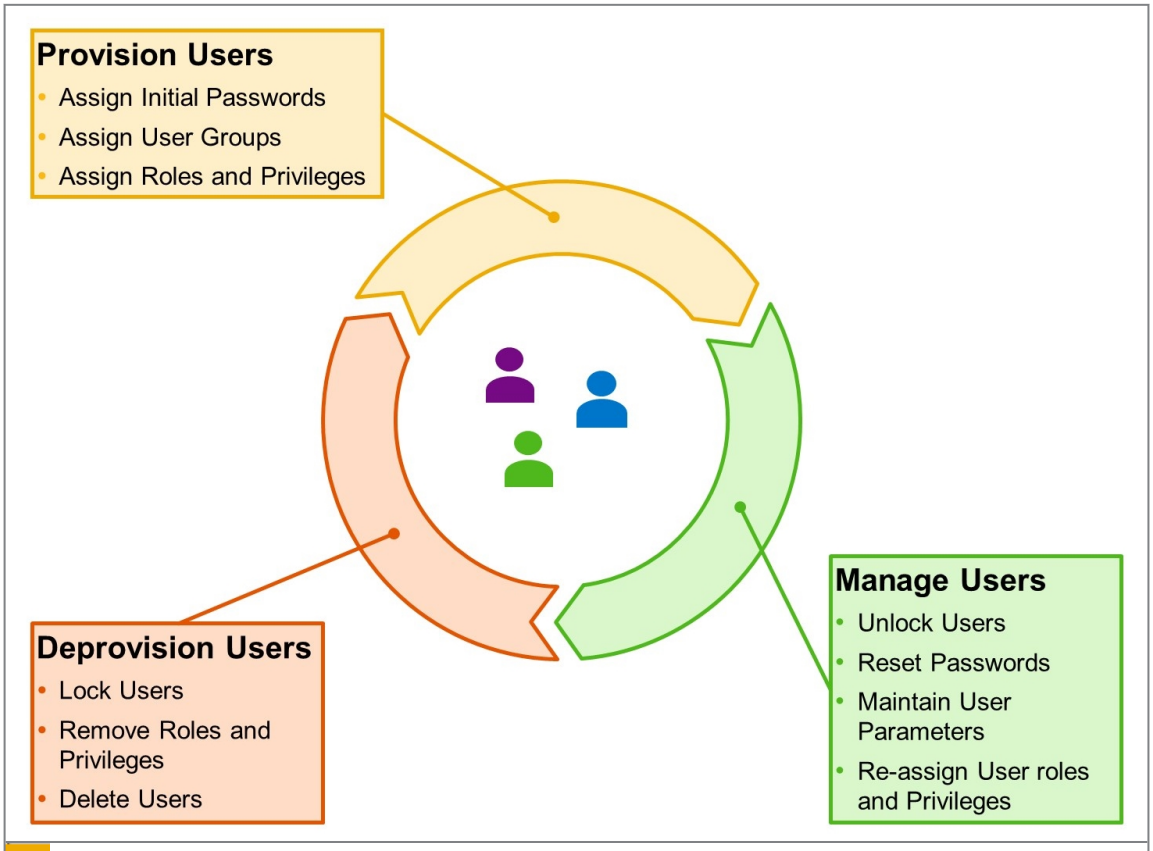


Figure 146: User Lifecycle Management

SAP HANA database users may be technical users or correspond to real end users. Several tools are available for user management.

The user administrator needs to create user account, grant privileges to the required database resources. In large deployments, the user administrator can divide the users in to groups and segregate the user administrator duties to user group administrators.

SAP HANA User Groups

User groups allow you to manage related users together. Group administrators can be assigned to manage individual user groups exclusively and independently of each other.

> Note:
> A user can belong to only one user group, but a user does not have to be a member of any group.

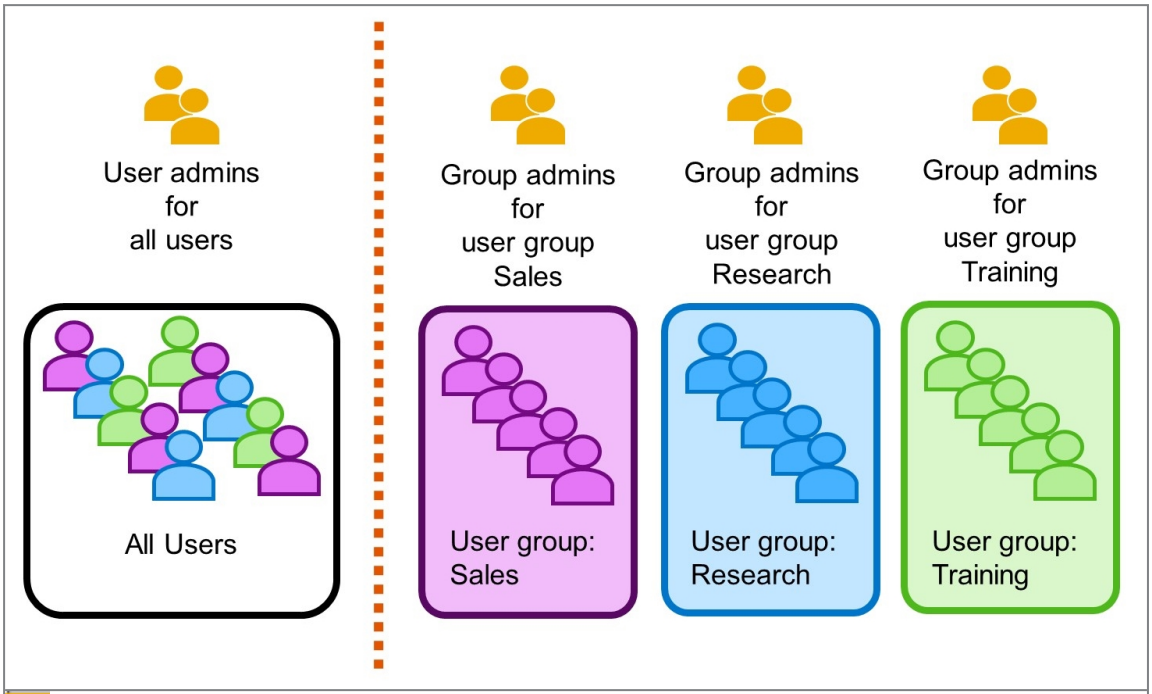


Figure 147: SAP HANA User Groups

Group Administration

A global user administrator (that is, a user with system privilege USER ADMIN) creates user groups. If the group requires its own exclusive administrator, the user administrator can configure this either when creating the group, or by editing the group later. The user administrator then designates one or more group administrators. A user can be a group administrator of more than one group.

If the user group is configured for exclusive administration, only the designated group administrator(s) can modify the group, for example, adding new users to the group or removing users from the group. In this way, groups of users can be managed independently of each other.

If a user group is not configured for exclusive administration, then both the global user administrator and group administrators can modify the group.

SAP HANA User Types

It is often necessary to specify different security policies for different types of database user. In the SAP HANA database, we differentiate between database users that correspond to real people and technical database users (interface users).

> Note:
> Technically, database users that correspond to real people and technical database users are the same. The only difference between them is conceptual.

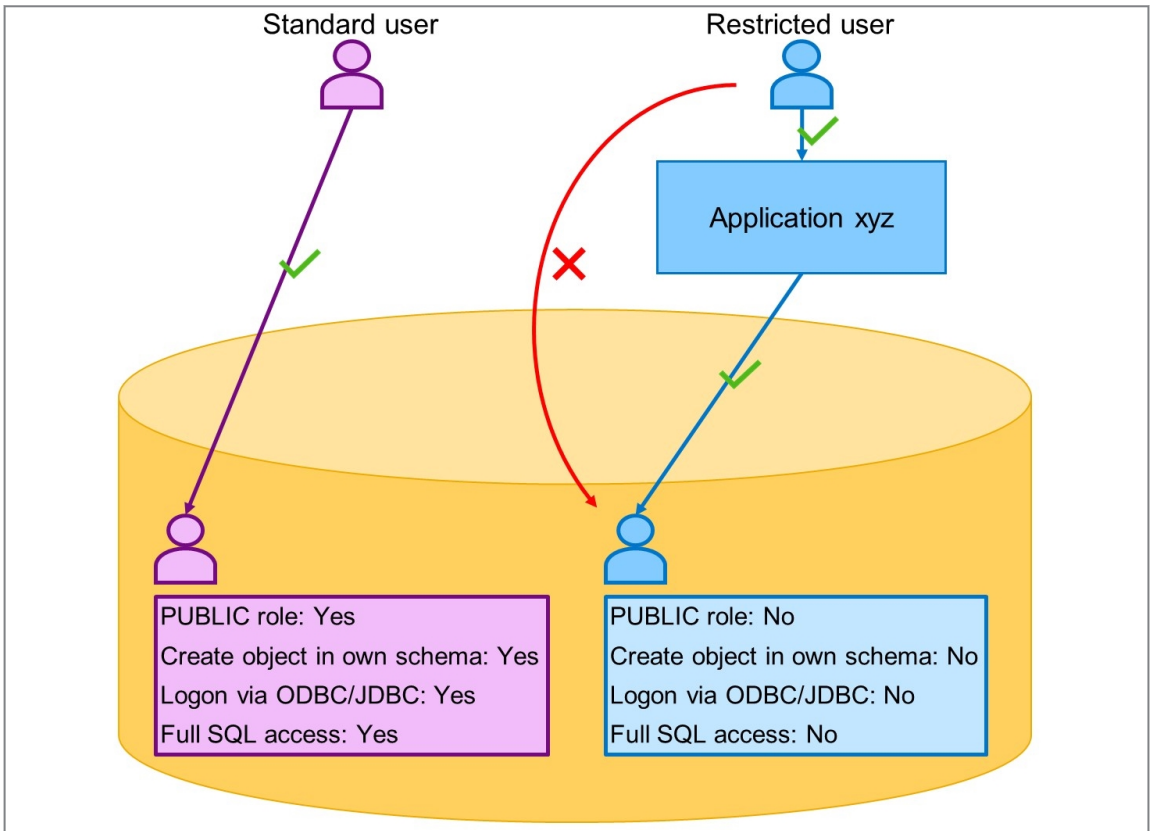


Figure 148: SAP HANA User Types

Database Users that Correspond to Real People

For every person who needs to work with SAP HANA, the user administrator creates a database user.

Database users that correspond to real people are dropped when the person leaves the organization. This means that any database objects that they own are also automatically dropped, and any privileges that they granted are automatically revoked.

Database users can be created with either the **`CREATE USER`** or **`CREATE RESTRICTED USER`** statement, or using the SAP HANA cockpit.

Standard Users

Standard users correspond to users created with the CREATE USER statement. By default they can create objects in their own schema and read data in system views. Read access to system views is granted by the PUBLIC role, which is granted to every standard user.

Restricted Users

Restricted users, which are created with the CREATE RESTRICTED USER statement, initially have no privileges. Restricted users are intended for provisioning users who access SAP HANA through client applications and who are not intended to have full SQL access via an SQL console. If the privileges required to use the application are encapsulated within an application-specific role, then it is necessary to grant the user only this role. In this way, it can be ensured that users have only those privileges that are essential to their work.

Compared to standard database users, restricted users are initially limited in the following ways:

- They cannot create objects in the database as they are not authorized to create objects in their own database schema.
- They cannot view any data in the database as they are not granted the standard PUBLIC role.
- They are only able to connect to the database using HTTP/HTTPS.

Technical Database Users

Technical database users do not correspond to real people. They are therefore not dropped if a person leaves the organization. This means that they should be used for administrative tasks such as creating objects and granting privileges for a particular application.

Some technical users are available as standard, for example, the users SYS and _SYS_REPO.

Other technical database users are created for application-specific purposes. For example, an application server may log on to the SAP HANA database using a dedicated technical database user.

SAP HANA Authorization Framework

When a user accesses the SAP HANA database using a client interface (for example, ODBC, JDBC, or HTTP), his or her ability to perform database operations on database objects is determined by the privileges that he or she has been granted.

All the privileges granted to a user, either directly or indirectly through roles, are combined. This means that whenever a user tries to access an object, the system performs an authorization check on the user, the user's roles, and directly granted privileges. It is not possible to explicitly deny privileges. This means that the system does not need to check all the user's privileges. As soon as all requested privileges have been found, the system skips further checks and grants access.

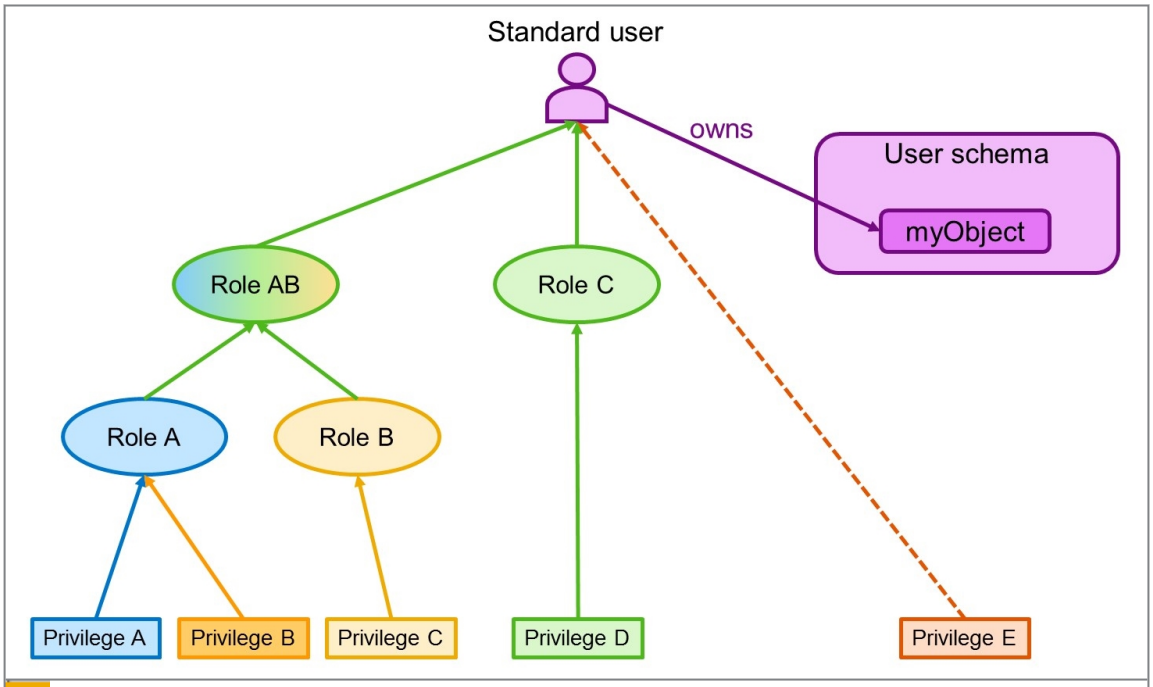


Figure 149: SAP HANA Authorization Framework

Privileges

Privileges allow users to work with database object, to perform actions, or to execute procedures in the SAP HANA database. SAP HANA uses several privilege types (system, object, analytic, package, and application) are used in SAP HANA.

- System privilege

  System privileges control general system activities. They are mainly used for administrative purposes, such as creating schemas, creating and changing users and roles, performing data backups, managing licenses, and so on.

  System privileges are also used to authorize basic repository operations.

- Object privilege

  Object privileges are used to allow access to and modification of database objects, such as tables and views. Depending on the object type, different actions can be authorized (for example, SELECT, CREATE ANY, ALTER, DROP, and so on).

  Schema privileges are object privileges that are used to allow access to and modification of schemas and the objects that they contain.

- Analytic privilege

  Analytic privileges are used to allow read access to data in SAP HANA information models (that is, analytic views, attribute views, and calculation views) depending on certain values or combinations of values. Analytic privileges are evaluated during query processing.

  Analytic privileges granted to users in a particular database authorize access to information models in that database only.

- Package privilege

  Package privileges are used to allow access to and the ability to work in packages in the classic repository of the SAP HANA database.

Packages contain design time versions of various objects, such as analytic views, attribute views, calculation views, and analytic privileges.

- Application privilege

  Developers of SAP HANA XS classic applications can create application privileges to authorize user and client access to their application. They apply in addition to other privileges, for example, object privileges on tables.

Database Roles

Privileges can be granted directly to users of the SAP HANA database. However, roles are the standard mechanism of granting privileges as they allow you to implement complex, reusable authorization concepts that can be modeled on business roles.

A database role is a collection of privileges that can be granted to either a database user or another role in runtime.

A role typically contains the privileges required for a particular function or task, for example:

- Business end users reading reports using client tools such as Microsoft Excel.
- Modelers creating models and reports.
- Database administrators operating and maintaining the database and its users.

Securing SAP HANA

SAP HANA Security Framework

The identity of database users accessing SAP HANA is verified through a process called authentication. SAP HANA supports several authentication mechanisms, several of which can be used for the integration of SAP HANA into single sign-on environments (SSO). The mechanisms used to authenticate individual users is specified as part of the user definition.

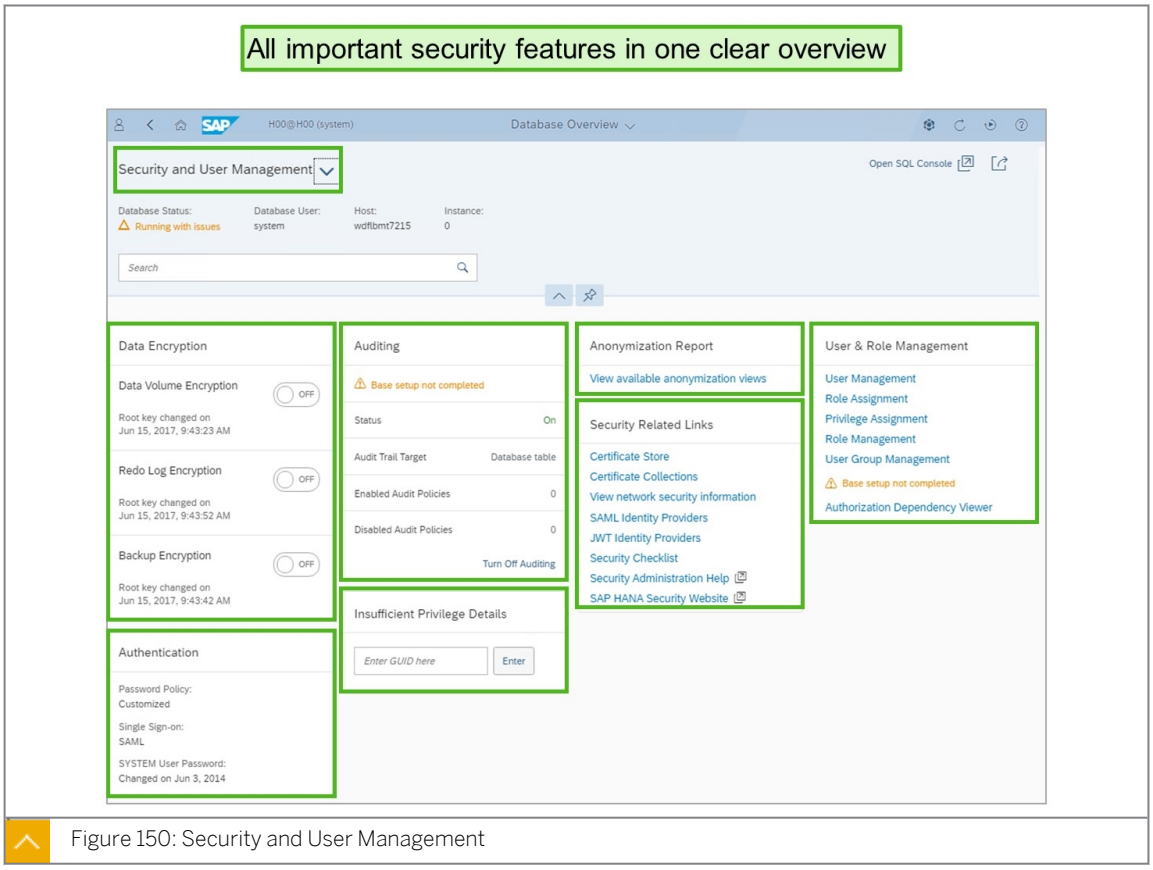


Figure 150: Security and User Management

User Authentication Mechanisms

Authentication mechanisms supported in SAP HANA. Mechanisms that are not required can be disabled.

Supported Authentication Mechanisms

- Basic authentication (user name and password)

  Users accessing the SAP HANA database authenticate themselves by entering their database user name and password.

- Kerberos, SPNEGO

  A Kerberos authentication provider can be used to authenticate users accessing SAP HANA Directly from ODBC and JDBC database clients, indirectly from front-end applications such as SAP BusinessObjects applications or via HTTP/HTTPS access by means of SAP HANA Extended Services (SAP HANA XS), classic mode.

- Security Assertion Markup Language (SAML)

  A SAML bearer assertion can be used to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients. SAP HANA can act as a service provider to authenticate users accessing via HTTP/HTTPS by means of SAP HANA XS classic.

- Logon and assertion tickets

  Users can be authenticated by SAP logon or assertion tickets issued to them when they log on to an SAP system that is configured to create tickets (for example, the SAP Web Application Server or Portal).

- X.509 client certificates

For HTTP/HTTPS access to SAP HANA by means of SAP HANA XS classic, users can be authenticated by client certificates signed by a trusted Certification Authority (CA), which can be stored in the SAP HANA XS trust store.

- JSON Web Token (JWT)

  A JSON Web Token can be used to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients or indirectly through SAP HANA extended application services, advanced model (SAP HANA XS, advanced).

Separate, database-specific authentication is possible for every certificate-based authentication mechanism since it is possible to create different certificate collections for individual purposes directly in every database, and every database can have its own key pair and public key certificate.

By default all authentication mechanisms are enabled, but it is possible and recommended to disable those that are not used in your environment.

Auditing Activity in SAP HANA Systems

Auditing provides you with visibility on who did what in the SAP HANA database (or tried to do what) and when. This allows you, for example, to log and monitor read access to sensitive data.

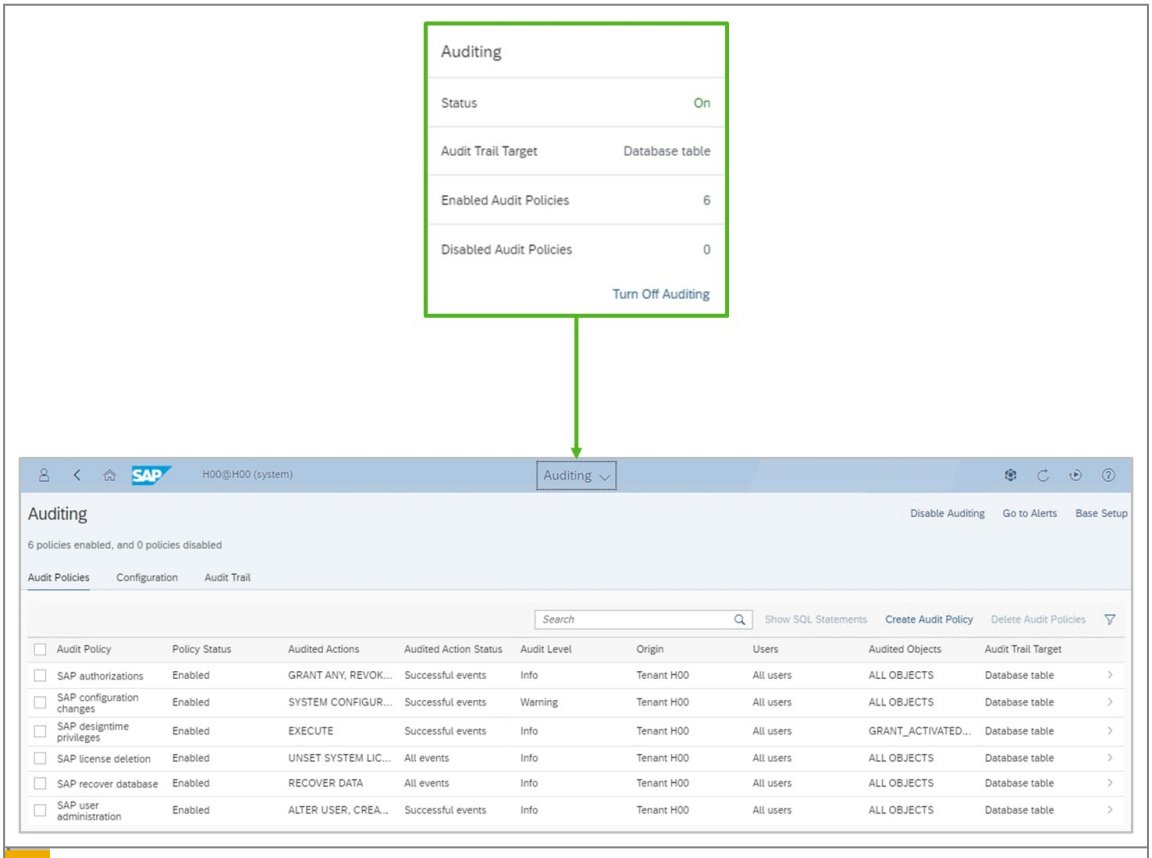


Figure 151: SAP HANA Auditing Application

Auditing allows you to monitor and record selected actions performed in the SAP HANA database. Auditing can be enabled individually and independently for every database in the system. Although auditing does not directly increase your system's security, if wisely designed, it can help you achieve greater security in the following ways:

- Uncover security holes if too many privileges were granted to some user
- Show attempts to breach security
- Protect the system owner against accusations of security violations and data misuse
- Allow the system owner to meet security standards

The following actions are typically audited:

- Changes to user authorization
- Creation or deletion of database objects
- Authentication of users
- Changes to system configuration
- Access to or changing of sensitive information

Only actions that take place inside the database engine can be audited. If the database engine is not online when an action occurs, it cannot be detected and therefore cannot be audited. These situations can arise during an upgrade of the database system, changing configuration files directly on operating system level and changing the password of the SYSTEM user by starting the name server in emergency mode.

Audit Policies

An audit policy defines the actions to be audited, as well as the conditions under which the action must be performed to be relevant for auditing. When an action occurs, the policy is triggered and an audit event is written to the audit trail. Audit policies are database specific.

Audit Trails

When an audit policy is triggered, that is, when an action in the policy occurs under the conditions defined in the policy, an audit entry is created in one or more audit trails.

Auditing Configuration and Audit Policy Management

To audit database activity, auditing must first be enabled in the database, and if necessary audit trails configured. It is then possible to create and activate the required audit policies. Audit policies can also be deactivated and reactivated later, or deleted altogether.

Auditing Configuration and Audit Policy Management can be setup using the SAP HANA Cockpit application Auditing an/or SQL via the SQL console.

Securing Internal and External Communication

The network communication channels used by SAP HANA can be categorized into those used for database clients connecting to SAP HANA and those used for internal database communication. SAP recommends using encrypted communication channels where possible.

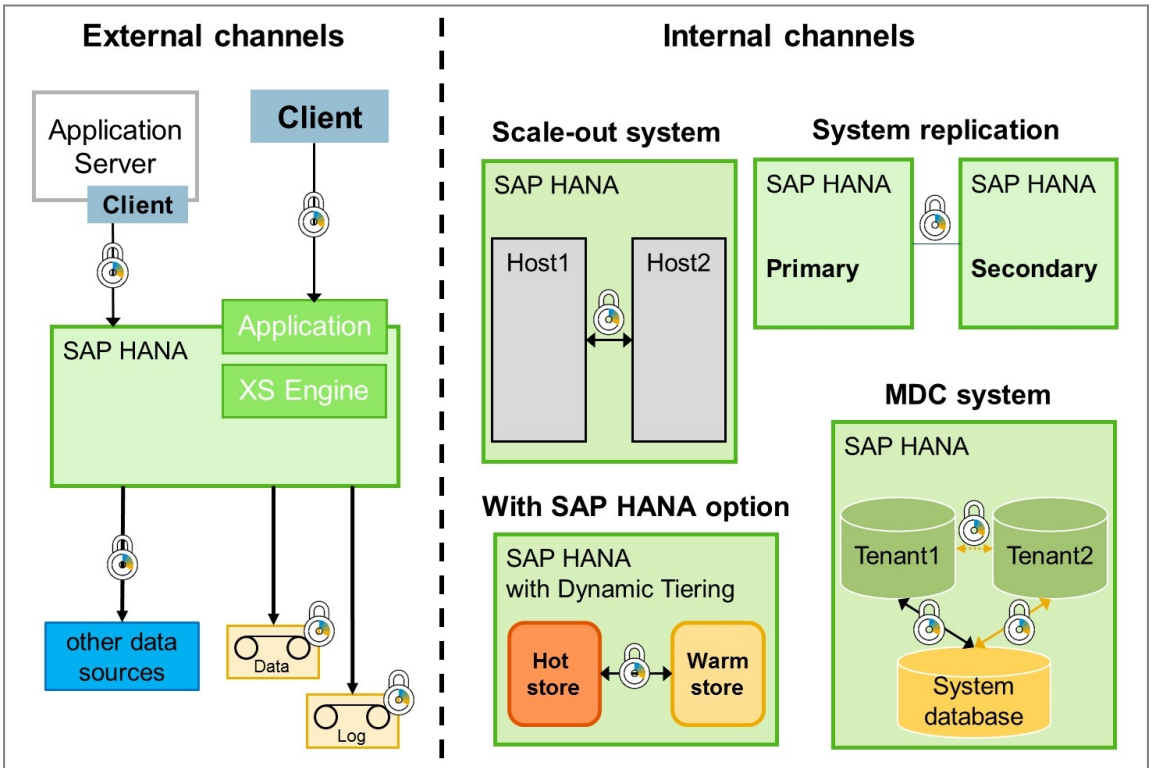


Figure 152: Securing Internal and External Communication

Internal Communication Channels

Channels used for SAP HANA internal communication within the database, between hosts in multiple-host systems, and between systems in system-replication scenarios.

Internal Communication Channels are:

- Internal database communication
- Internal communication between hosts in a distributed (multi-host) SAP HANA system
- Internal communication between systems at the different sites in a system replication (high availability) scenario
- Internal communication between the SAP HANA database and server components, such as extended storage (SAP HANA dynamic tiering)

External Communication Channels

Channels used for external access to SAP HANA functionality by end-user clients, administration clients, application servers, and for data provisioning through SQL or HTTP.

- Connections used for administrative purposes.
- Connections used for data provisioning.
- Connections from database clients that access the SQL/MDXinterface of the SAP HANA database.
- Connections from HTTP(S) clients.
- Outgoing connections.

Data Storage Security in SAP HANA

Several mechanisms can be used to protect security-relevant data used by the SAP HANA database from unauthorized access. All passwords in SAP HANA are stored securely in a hashed and salted form and never in clear text.

Data and Log Volume Encryption

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer. Data volume encryption protects the data area on disk, while redo log encryption protects the log area on disk.

Backup Encryption

SAP HANA supports native backup encryption. Backup encryption safeguards the privacy of the SAP HANA business data by preventing unauthorized parties from reading the content of backups. Backup encryption can be enabled for all backup types.

Password Policy

Passwords for the basic authentication of database users are subject to certain rules. These are defined in the password policy. You can change the default password policy in line with your organization's security requirements.

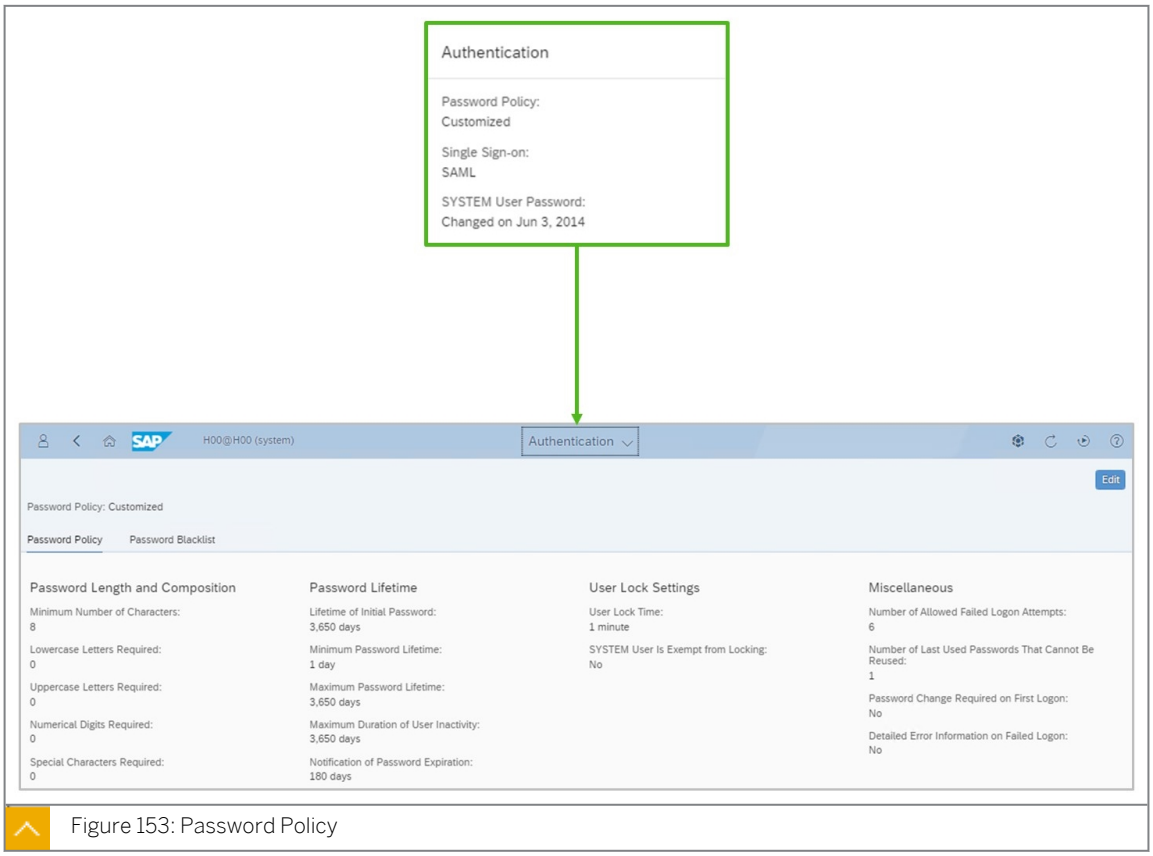


Figure 153: Password Policy

The password policy is defined by parameters in the password policy section of the indexserver.ini configuration file for tenant databases and the nameserver.ini configuration file for the system database. Although you can configure your password policy directly in the configuration file, we recommend that you use the SAP HANA cockpit.

Password Blacklist

A password blacklist list is a list of words that are not allowed as passwords or parts of passwords.

The password blacklist in SAP HANA is implemented with the table _SYS_PASSWORD_BLACKLIST in the schema _SYS_SECURITY. This table is empty when you create a new instance.

You can enter words in the password blacklist as part of password policy configuration on the Password Policy and Blacklist page of the SAP HANA cockpit.

The password blacklist can be managed for each database individually.

LESSON SUMMARY

You should now be able to:

- Understanding Administration Security

Learning Assessment

1. Which file system files can be encrypted by SAP HANA?

   Choose the correct answers.

   - ☐ A Data Volumes
   - ☐ B Linux kernel configuration files
   - ☐ C Log Volumes
   - ☐ D Configuration Ini files
   - ☐ E Log Backups

2. Which of the following statements about the SAP HANA Authorization Framework are correct:

   Choose the correct answers.

   - ☐ A User can be assigned to groups.
   - ☐ B Privileges can be assigned to groups.
   - ☐ C Roles can be assigned to roles.
   - ☐ D Privileges can be assigned to users.

Learning Assessment - Answers

1. Which file system files can be encrypted by SAP HANA?

   Choose the correct answers.

   - [X] A Data Volumes
   - [ ] B Linux kernel configuration files
   - [X] C Log Volumes
   - [ ] D Configuration Ini files
   - [X] E Log Backups

   Indeed. SAP HANA can encrypt data Volumes, log volumes and data and log backups, but the configuration Ini files and the Linux kernel files are not.

2. Which of the following statements about the SAP HANA Authorization Framework are correct:

   Choose the correct answers.

   - [X] A User can be assigned to groups.
   - [ ] B Privileges can be assigned to groups.
   - [X] C Roles can be assigned to roles.
   - [X] D Privileges can be assigned to users.

   Correct. Privileges can't be assigned to groups.

UNIT 10 Migration to SAP HANA

Lesson 1

UNIT OBJECTIVES

- Describe the Key Steps in a Migration to SAP HANA DB

Unit 10
Lesson 1

Key Steps in a Migration to SAP HANA DB

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe the Key Steps in a Migration to SAP HANA DB

Description of the Key Steps of a Migration to SAP HANA

We know that more and more applications developed by SAP are built to run only on the SAP HANA database, such as SAP S/4HANA. Many customers are migrating from their legacy applications to the next generation applications that run only on SAP HANA. In this case, a migration of their application and also the database is required. But there are customers who want to stay with their legacy application but swap the database to SAP HANA to take advantage of improved performance. In this case, only the database is migrated.

SAP provide tools to migrate your legacy application to next generation application and also the database to SAP HANA. The migration of the application and the database can also be performed with one tool in one step.

A key question that determines the best tool to use is: what is in the scope of the migration?

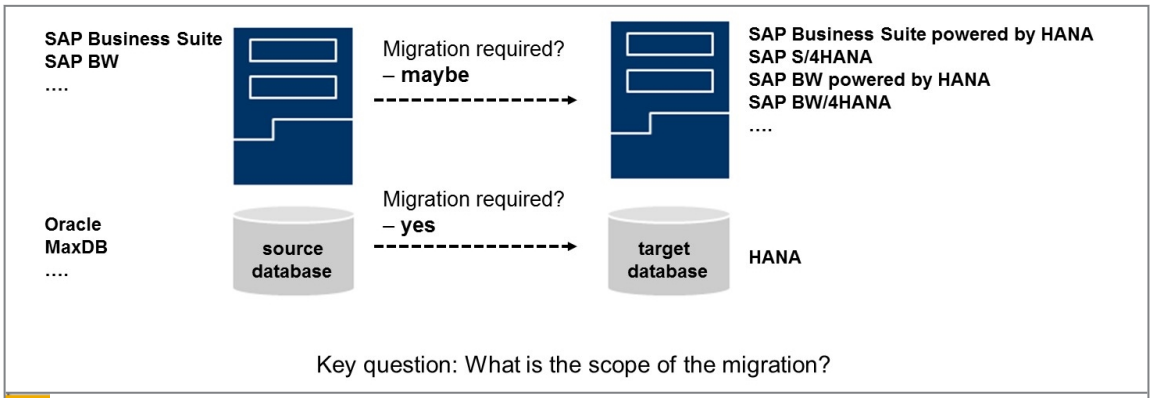


Figure 154: What is the End Goal?

To migrate to SAP HANA database, there are three main migration options as follows:

Possible Options

- Option 1 - New installation of an SAP application that runs on SAP HANA.
- Option 2 - Classical two-step migration of first, the application and then the database.
- Option 3 - Simultaneous update of an existing application and also a migration to the SAP HANA database.

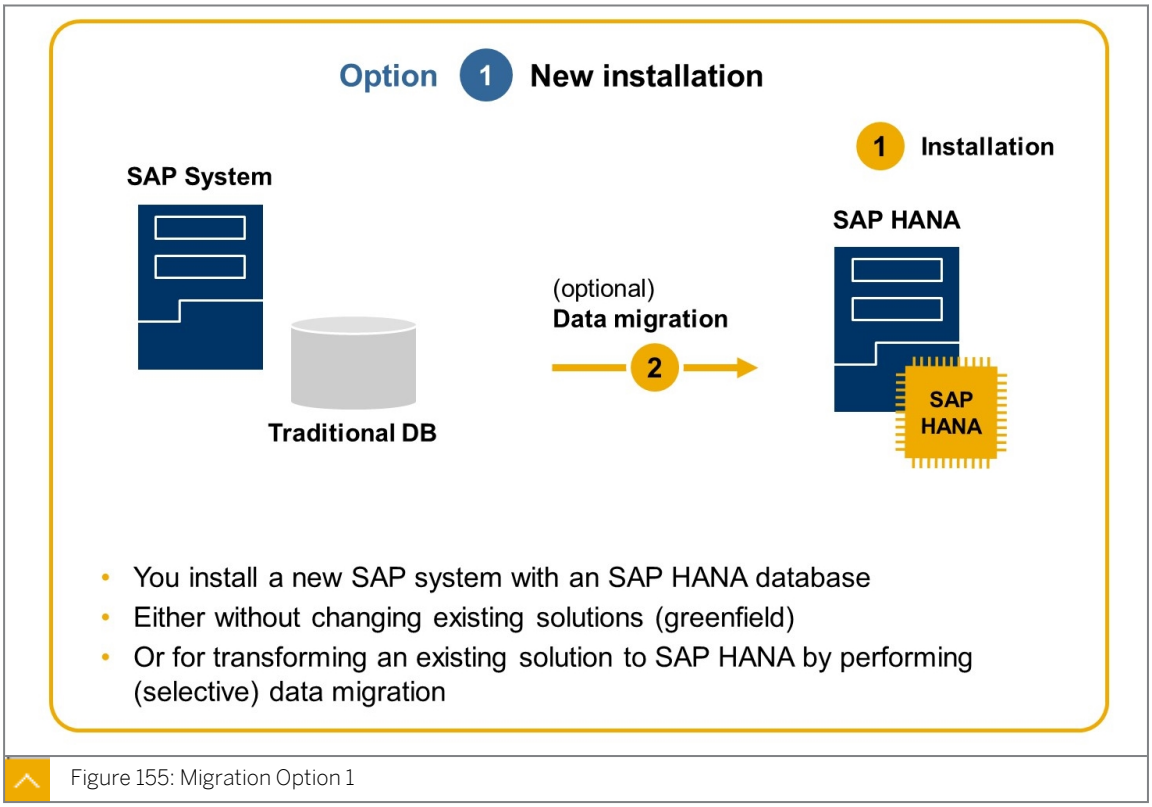


Figure 155: Migration Option 1

Option 1 is a new installment of an SAP application that runs on SAP HANA (referred to as greenfield). The option allows for the addressing of changes to an existing solution landscape. For example, by migrating only the data that is required and leaving old data behind. Or by transporting only development content such as BW queries, or custom ABAP reports. Various tools can help with the migration of data and content objects such as SAP Landscape Transformation Replication Server for a table level data migration, or using ABAP transports for content object migration. This might also be an opportunity to consolidate a fragmented landscape or even the other way: splitting up a system into several target systems. There are tools to manage all aspects and choices in this type of scenario. One of the key benefits of this approach is that you can continue to run your legacy system in parallel and decommission only when you are ready. The downside is that you need additional hardware and resources to support the new installation.

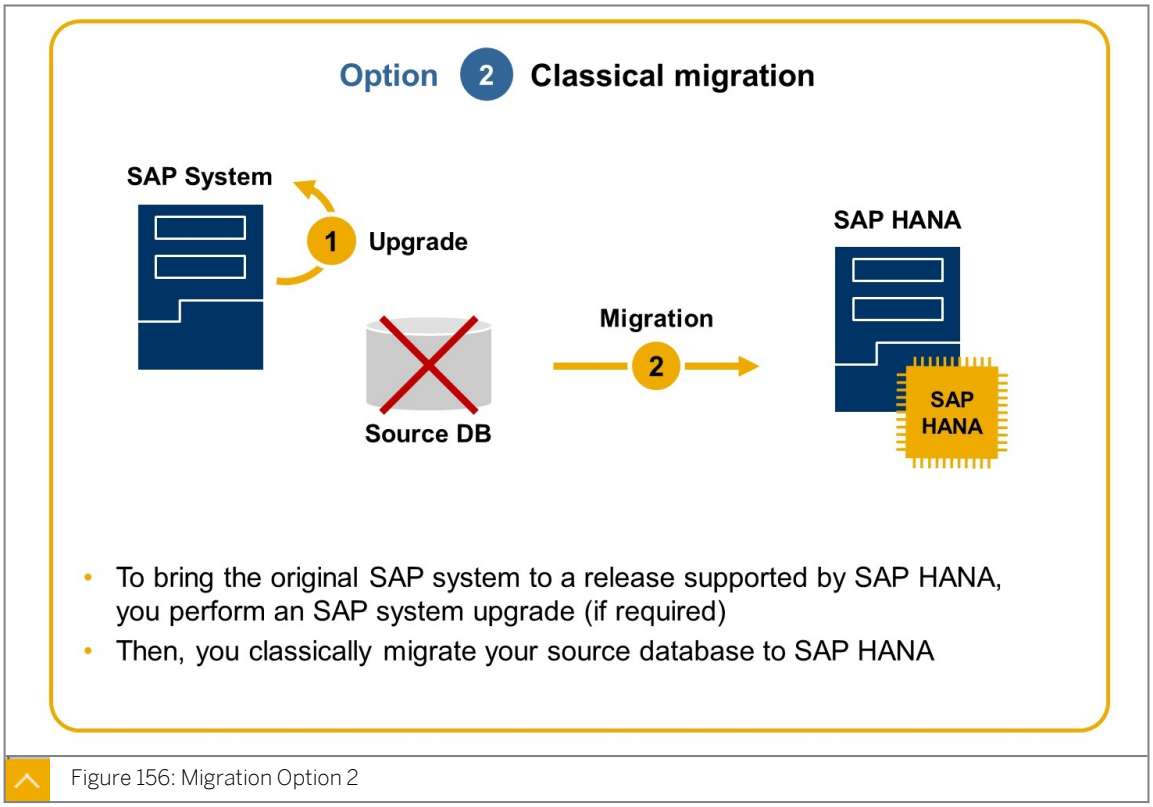


Figure 156: Migration Option 2

Option 2 first brings the SAP application to a release supported by SAP HANA if this is required, by performing an SAP application upgrade with Software Update Manager (SUM). Afterwards, the database is migrated to SAP HANA, using Software Provisioning Manager (SWPM).

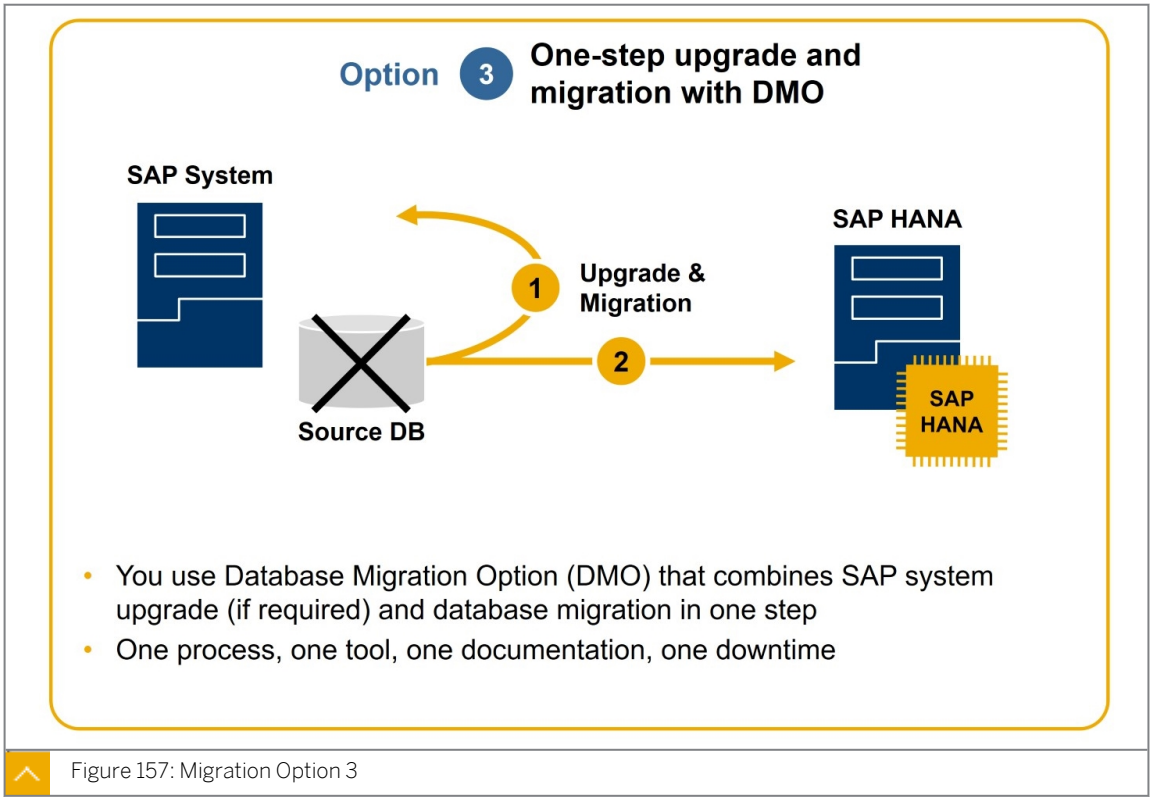


Figure 157: Migration Option 3

Option 3 is the DMO approach. Here, SUM performs the SAP application upgrade and then migrates the database **all in one step**. DMO is not a tool, but a selectable option within the SUM tool. There are many options within the SUM tool that you can choose to define the steps of a migration.

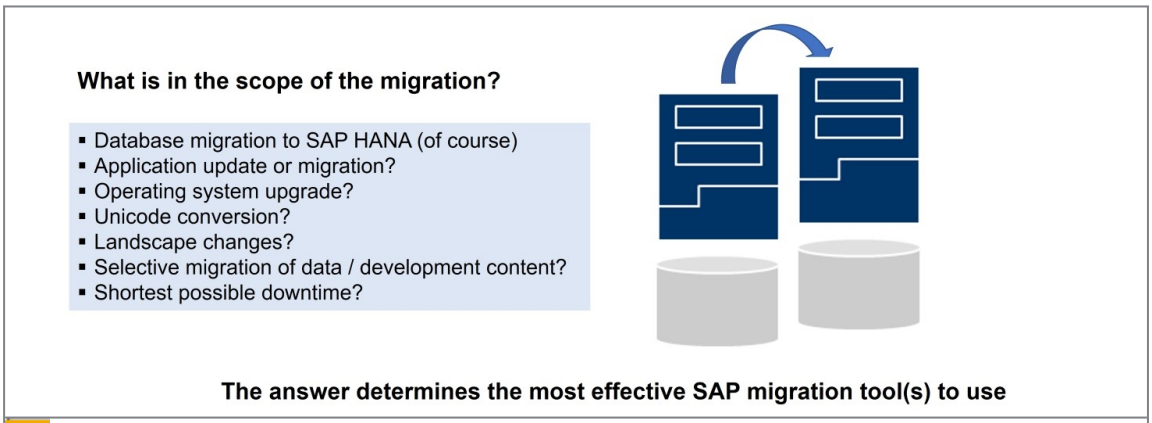


Figure 158: Scope of Migration

A migration is potentially a complex process involving many steps. If you are combining a migration of the application you might need to also perform mandatory steps such as applying patches / support packs, Unicode conversion and so on.

Some customers may chose to perform these steps one by one, whereas others may prefer an all-in-one approach. This is where a thorough investigation of the tools is required.

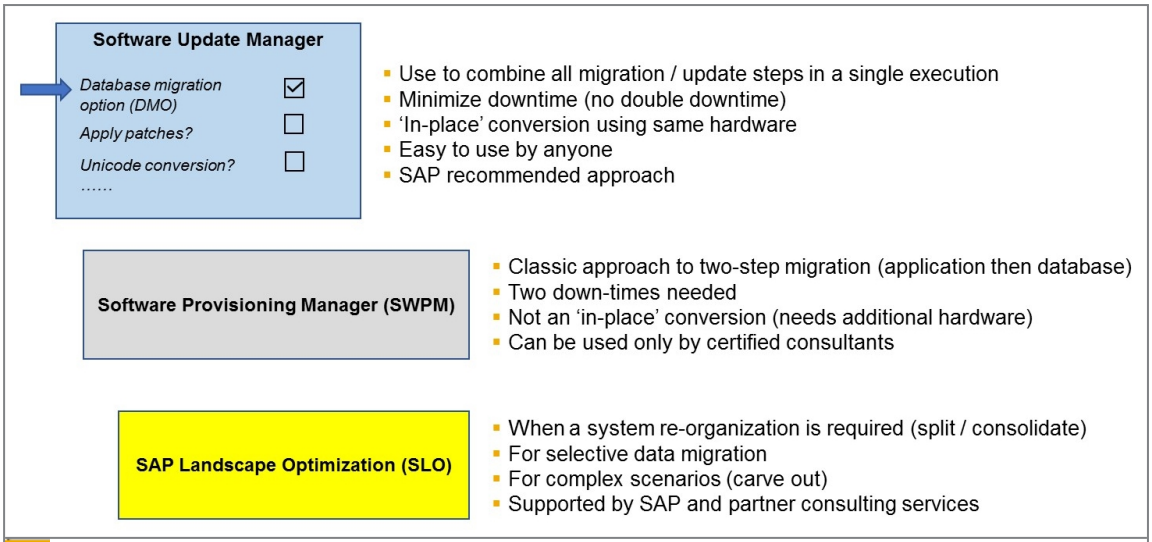


Figure 159: Migration Tools

There are three key tools, or tools sets to support a migration to SAP HANA:

1. Software Update Manager (using the Database Migration Option)
2. Software Provisioning Manager
3. SAP Landscape Optimization

In most cases, the DMO of SUM approach is recommended. But for various reasons, an all in one step might not be possible and so a classical migration would be an alternative using SWPM. For landscape re-organizations due to mergers and acquisitions SAP and partners

can support using the toolkit of SAP Landscape Optimization (SLO) to develop a new optimal landscape running on SAP HANA.

In special cases, if no migration is desired or possible, a new installation would be the possible exception.

LESSON SUMMARY

You should now be able to:

- Describe the Key Steps in a Migration to SAP HANA DB

Learning Assessment

1. What are the three tools that might be used in a migration to SAP HANA?

   Choose the correct answers.

   - ☐ A Database Migration Option of the Software Update Manager
   - ☐ B Software Provisioning Manager
   - ☐ C SAP Landscape Replication Server
   - ☐ D SAP HANA Migration Assistant

Learning Assessment - Answers

1. What are the three tools that might be used in a migration to SAP HANA?

   Choose the correct answers.

   - [X] A Database Migration Option of the Software Update Manager
   - [X] B Software Provisioning Manager
   - [X] C SAP Landscape Replication Server
   - [ ] D SAP HANA Migration Assistant

   Correct! Only SAP HANA Migration Assistant is not a tool used in a database migration to SAP HANA, this is a tool used to migrate HANA development content from HANA XS to HANA XS Advanced.

UNIT 11 Appendix

UNIT OBJECTIVES

- Develop your SAP HANA skills
- SAP HANA Glossary

Unit 11
Lesson 1

Develop Your SAP HANA Skills

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 skills

SAP HANA Learning Journeys

This course provided a basic introduction to all aspects of SAP HANA. To move to the next level, you need to consider your role and the skills you will need to develop to perform your duties. SAP offers many learning resources including training courses and webinars. To prove your skills in the market, you can also take the official SAP Certification to prove you have developed the skills to perform effectively in your role.

To make it easier for you to find the learning resources that you need, SAP has developed **Learning Journeys**.

Learning Journeys are structured visual guides, designed to help you understand and navigate the path to become fully competent with any SAP solution.

For SAP HANA, there are four Learning Journeys. On the following diagram, we include QR codes that will take you directly to each SAP Learning Journey.

Note:
The SAP Learning Journey's are updated from time-to-time as new learning assets become available. The screen-shots below are used only as an illustration. Always go online and check the latest SAP Learning Journeys for the update information on course recommendations and sequences. Use the QR codes below or the link https://help.sap.com/ to explore the latest SAP Learning Journeys.

SAP HANA Developer

This Learning Journey covers the role of an application developer who will code using ABAP on HANA (for example BW on HANA, Suite on HANA, S/4HANA) and also native HANA development using languages such as Java, Node.JS.

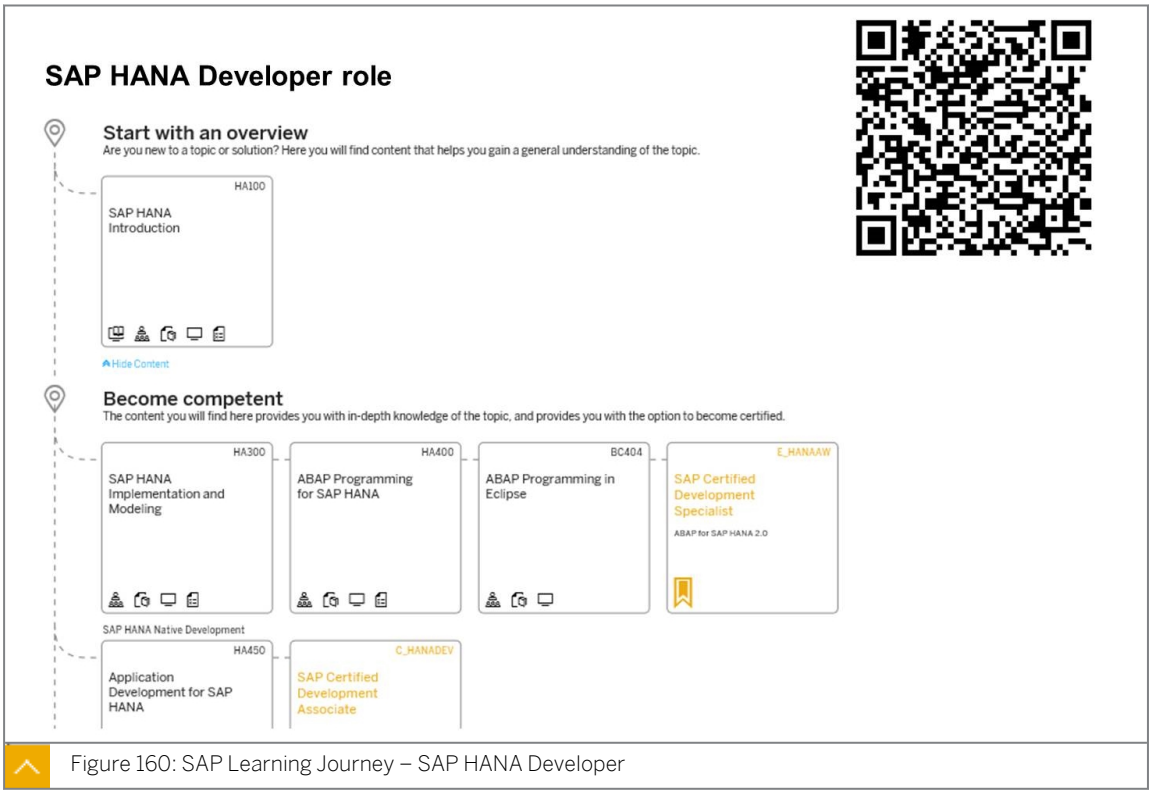


Figure 160: SAP Learning Journey – SAP HANA Developer

SAP Learning Journey — SAP HANA Administrator

This Learning Journey covers the role of an administrator or support consultant who will install and monitor SAP HANA and also perform routine operations such as backups and user maintenance.

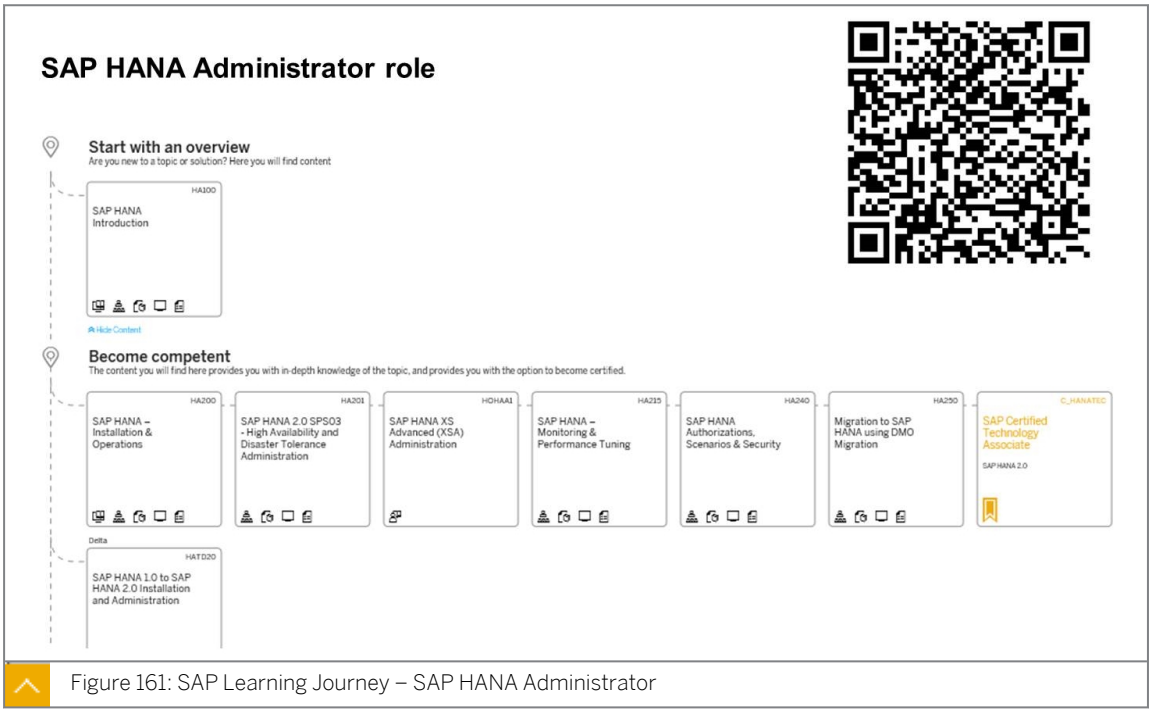


Figure 161: SAP Learning Journey – SAP HANA Administrator

SAP Learning Journey – SAP HANA Modeler

This Learning Journey covers the role of a modeler who is responsible for the development and maintenance of modeling artifacts such as calculation views and functions which are used to source data to analytical applications. This is the best journey for a business intelligence consultant/data scientist.

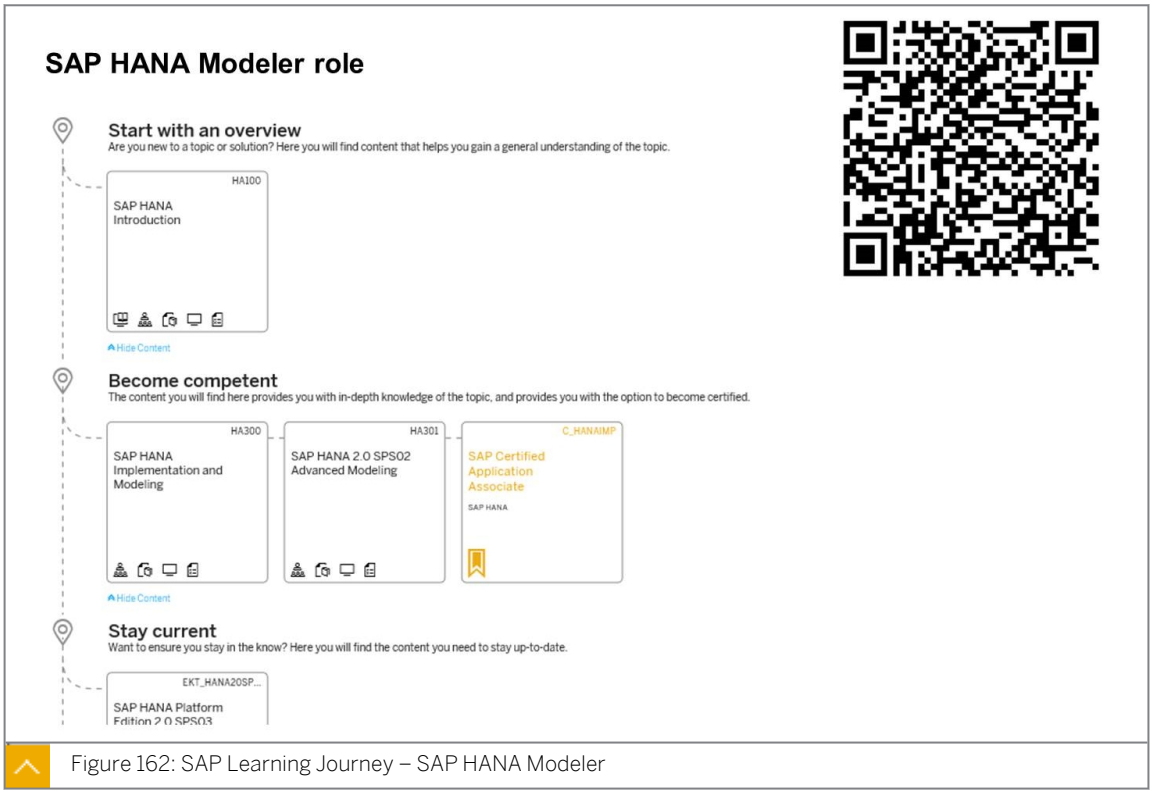


Figure 162: SAP Learning Journey – SAP HANA Modeler

SAP Learning Journey – SAP HANA Data Provisioning

This Learning Journey covers the role of a consultant who is responsible for provisioning data to SAP HANA using the various replication, ETL, and streaming tools. Quite often this role is taken on by any of the other roles so it may be applicable to your skills development. Check with your project manager to find out who will develop data loading scenarios using SDI/SDQ and other tools.

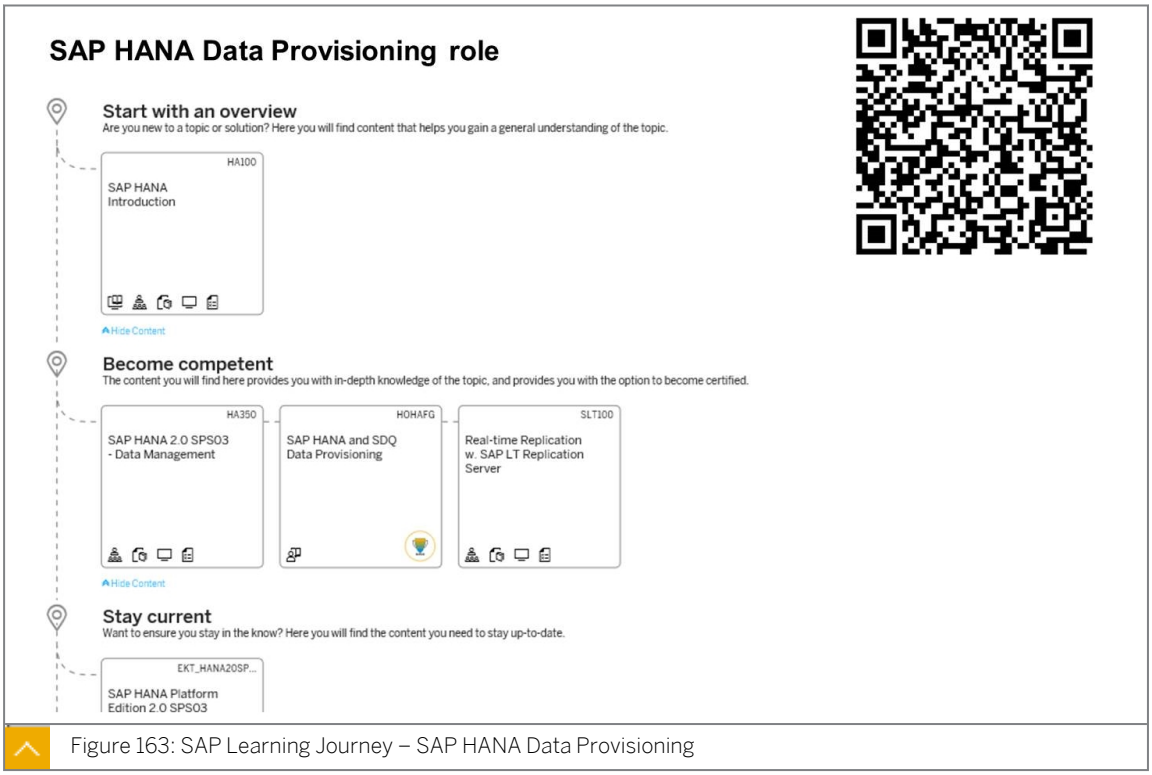


Figure 163: SAP Learning Journey – SAP HANA Data Provisioning

LESSON SUMMARY

You should now be able to:

- Develop your SAP HANA skills

Unit 11
Lesson 2

Glossary

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

- SAP HANA Glossary

LESSON SUMMARY
You should now be able to:

- SAP HANA Glossary

Glossary

Active/Active-Read Enabled Mode
Use of a secondary, redundant server to handle read only, intensive queries thereby allowing the primary server to focus on read/write activities

AFL
Application Function Library is a repository of SAP supplied, ready made modules that can be used to build predictive analysis models and business applications

Analytic Privilege
Object to define the data access rules for users, e.g. company code = 1000

Analytic View
Modeling object used to define a star schema based data model, but is now replaced by the calculation view.

AnyDB
Is the way SAP refers to all other database systems not being SAP HANA.

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.

BWoH
Unofficial term for SAP BW powered by HANA

Cache
Data storage mechanism made up of hardware and software that holds frequently used data so that future requests can be server faster.

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.

Card (SAP Fiori)
Each task or topic on an SAP Fiori Overview page is represented by a card.

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.

CLI
A Command Line Interface (CLI) is a text-based user interface often used by Linux system administrators to view and manage computer files.

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.

Cockpit (SAP HANA Cockpit)
An interface used by administrators and support staff to monitor SAP HANA and perform routine operations to the database such as backups and user creation.

Computing node
SAP HANA computing node are the hosts that hold the database data and perform the query execution on that data. A computing node is also often referred to as worker node.

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 and is one of the types of calculation view.

Data Warehouse Foundation (DWF)
Toolset that can be installed in SAP HANA to provide developers with tools to build an SQL data warehouse completely on SAP HANA without the need for BW.

Database Explorer
A tool of Web IDE for SAP HANA that provides developers with direct access to the database of SAP HANA.

Delivery Unit
Used to transport development artifacts from one HANA system to another, used only in classic XS developments.

Delta Merge
A essential periodic database operation that merges newly inserted uncompressed data of a column table to the main store where existing compressed data resides.

Dimension
Collection of attributes that belong together (for example, product color, weight, price) used to provide meaning to measures and can be shared across many calculation views.

DMO
Database Migration Option (DMO), an option of the Software Update Manager (SUM) tool and is chosen when a database migration (to HANA) is required as well as an update of the application all in one step.

DRAM
Dynamic RAM is a type of computer memory that offer the fastest data access and is used by HANA for the storage of hot data, but is volatile and so data is lost when power is gone.

Dynamic Tiering
See Extended Storage

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. Also known as Dynamic Tiering.

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)

HaaS
HANA as a Service — access SAP HANA as a public cloud service

hdbclient
SAP HANA provides client interfaces for connecting applications as part of the SAP HANA client software package. The SAP HANA client can be installed on both Unix/Linux and Microsoft Windows operating systems, as well as on an SAP HANA server host during server installation.

HDBLCM tools
The SAP HANA database lifecycle management (HDBLCM) tools are used to perform SAP HANA platform lifecycle management (LCM) tasks, including installing, updating, and configuring an SAP HANA system.

HDI
HANA Deployment Infrastructure — The deployment infrastructure used to build XSA applications, where an ‘all or nothing’ build approach applies

Linux
Linux is a family of open-source Unix-like operating systems based on the Linux kernel.

Linux is typically packaged in a Linux distribution like SUSE, Red Hat or Debian.

SAP HANA is supported on SUSE Linux Enterprise Server and Red Hat Enterprise Linux. See SAP Note

2235581: "SAP HANA Supported Operating Systems" for the specific details.

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.

Multi-store table
A table that is partitioned across memory and disk layers.

Multi-tenancy
An architecture that supports the complete isolation of applications running on the same platform.

Native DataStore Object (NDSO)
A component of DWF an NDSO is a persistence object used to store data in an SQL driven data warehouse built on SAP HANA, and provides data load request logging and delta management.

NSE
Native Storage Extension is a warm data tiering technology where data resides on disk and is loaded to memory as required using intelligent caching.

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.

PMEM
Persistent Memory is a type of computer memory that works alongside DRAM to provide a lower cost of implementing large memory with access that is fast, (but not as fast as DRAM) but PMEM is non-volatile and so data is not lost when power is gone.

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.

R
A popular open-source programming language used to develop statistical / predictive models that is integrated with SAP HANA.

Revisions (SAP HANA Revisions)
SAP HANA Revisions are bug fixing packages available for the SAP HANA components. Revisions are released inbetween SAP HANA Support Package Stacks.

Role

A collection of privileges that can be assigned to another role or a user

SAP BW/4HANA
SAP's next generation BW that is built to run only on SAP HANA.

SAP S/4HANA
SAP's next generation ERP suite that is built to run only on SAP HANA

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.

Savepoint
Savepoints are required to synchronize changes in memory with the persistency on disk level. All modified pages of row and column store are written to disk during a savepoint.

The data belonging to a savepoint represents a consistent state of the data on disk and remains untouched until the next savepoint operation has been completed.

Scale-out
Install SAP HANA across multiple servers to share the processing and data storage, and maybe also to provide redundant standby servers to be used in case of primary server failure. A scale-out system is also often referred to as distributed system or Multiple-host system.

Scale-up
Add more CPU and/or RAM to improve performance of SAP HANA .

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.

Series Data
Data that is collected at measurable intervals such as time

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.

SoH
Unofficial term for SAP Business Suite powered by HANA

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 (SPS) is a two-digit number that identifies the release of SAP HANA, for example, SAP HANA 2.0 SPS05.

SQL
Structured Query Language is used to read and write from relational databases. It is the world's most popular language for doing this.

SQLScript
The native database query language of SAP HANA that is an extension of standard SQL with additional support for column store, in-memory processing plus advanced data processing such as text, spatial

SSH
Secure Shell (SSH), is a remote administration protocol that allows Linux system administrator to securely administrate, control and modify remote servers over the network.

Standby server
A standby server is a passive SAP HANA node that is added to a SAP HANA scale-out system. When there is a server failure, the standby server will

take-on the role of the failed server and become a active member of the SAP HANA scale-out system.

Studio (SAP HANA Studio)
Interface used by developers and modelers and administrators for building XS data models and applications.

SUM
Software Update Manager (SUM), tool for maintaining SAP systems, e.g. applying SAP Support Packages, performing an SAP system upgrade, migrating the database, converting an SAP ECC system to SAP S/4HANA Server.

SWPM
Software Provisioning Manager is a tool used to perform a classic, two-step migration

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.

TensorFlow
Machine learning models provided by Google that can be accessed by SAP HANA

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.

Worker node
See Computing node.

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.