Capgemini

SAP HANA

Lesson Name: Introduction to SAP HANA

Contents

Lesson Objectives

After completing this lesson, participants will be able to -

- Know about SAP HANA
- Understand HANA Architecture
- Understand SAP In-Memory Concept
- Understand HANA Database Concepts
- Know about Database tables -
  - Row Store
  - Column Store
- Understand Data Provisioning

Introduction to SAP HANA

- SAP stands for System, Application, Product in Data Processing.
- HANA stands for High-performance Analytic Appliance.
- SAP HANA is a flexible, data-source-agnostic appliance that enables customers to analyze large volumes of SAP-ERP data in real-time.
- It avoids the need to materialize transformations.

Introduction to SAP HANA

SAP HANA appliance software is a hardware and software combination that integrates a number of SAP components.

- Components include in memory SAP HANA Database, Replication Server, Session Management , disk storage, persistence layer, relational engines and responds to SQL queries.
- SAP HANA database consists of multiple servers and the most important component is the Index Server. SAP HANA database consists of Index Server, Name Server, Statistics Server, Preprocessor Server and XS Server(Built-in application server). From release SPS11, SAP delivers more powerful XS server which is called XS Advance.
- SAP HANA Database is a hybrid in-memory database that combines row-based, column-based, and object-based database technology.

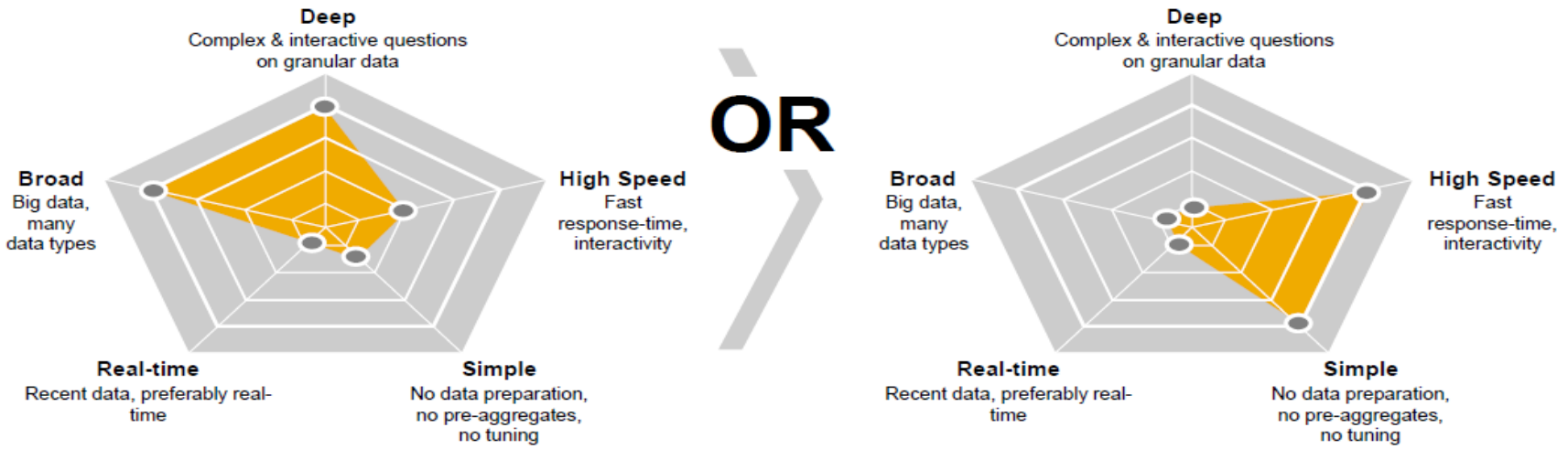
Introduction to SAP HANA

HANA is a Breakthrough Technology

- SAP HANA is a SQL database where all data is stored in-memory. Of course, it is much more complex:
  - HANA is sold as an appliance with dedicated hardware (provided by IBM, HP, Fujitsu, Cisco, Dell and more recently Hitachi)
  - It contains Solid State Disks (SSD) to store the updating logs (to respect ACID properties)
  - All data is regularly saved to classic Hard Disk Drives (HDD) to allow a reboot in case of a power cut for instance
  - The HANA license includes a modeling tool
  - The HANA license includes a loading or a replication tool.

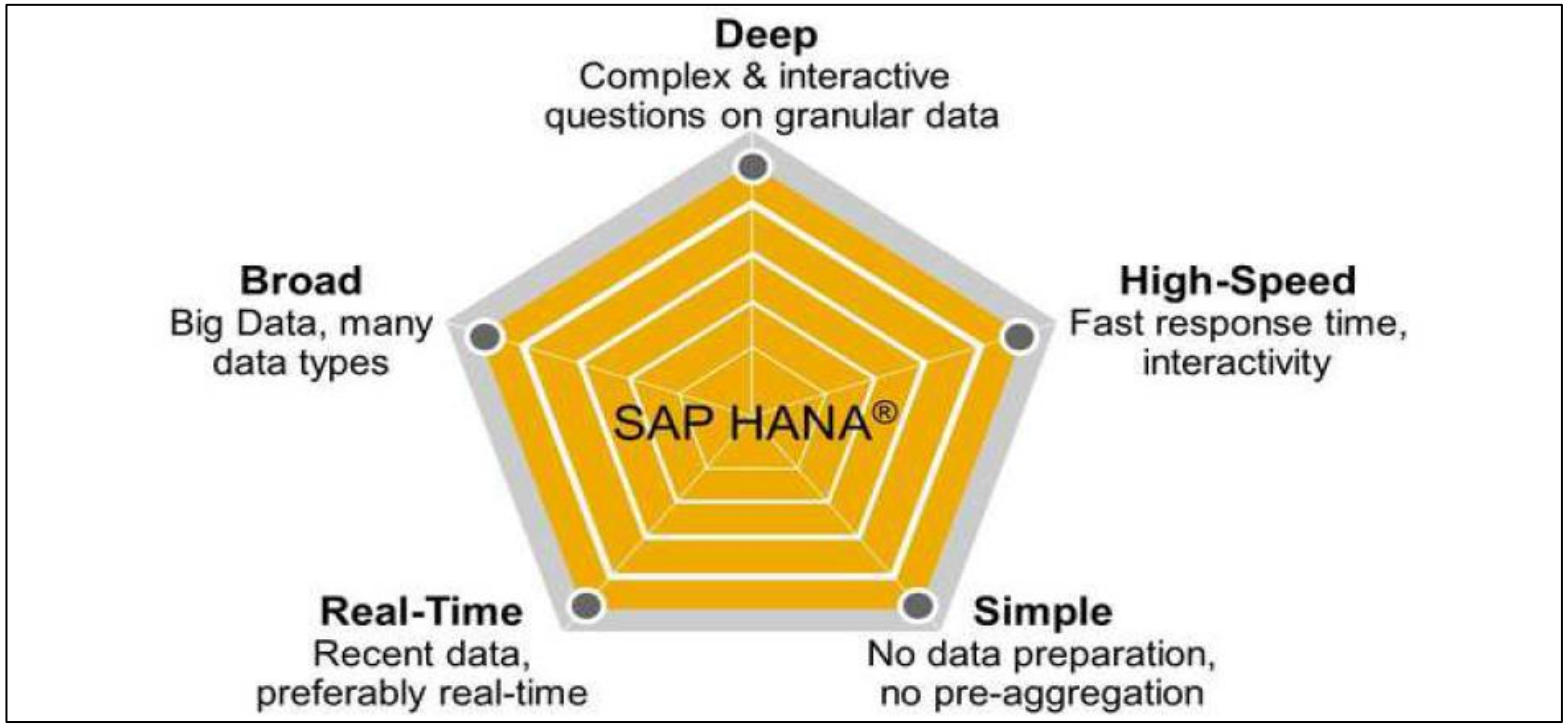
Introduction to SAP HANA

Need a breakthrough technology
Today's technology requires tradeoff



Before HANA, developing applications required several trade of decisions. Picture above, shows five typical dimensions while building a business applications. As an example, we may have to decide between providing a high-speed application and developing an application which does not required special preparation of processing data e.g. calculating aggregations beforehand and sorting the result set. Also it was very difficult to develop an application which is both real-time and able to analyze a large amount of data.

Introduction to SAP HANA



SAP HANA tries to deliver across all these five dimensions, so that you no longer have to make so many trade-off decisions. The goal is to enable the development of applications that combine Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) usage patterns.

Introduction to SAP HANA

Break-through Technology:

- The application developed so far will acknowledge the fact that this typically requires making several trade-off decisions.
- Trade Off is a situation that involves losing one quality or aspect of something in return for gaining another quality or aspect. If one thing increases, some other thing must decrease
- The graphics on the slide show the five dimensions of requirement that are typically required for a business applications.
- With a traditional database, addressing these all 5 dimensions have been conflicting so far and challenged developers very badly.
- Development of an application which is both real-time and able to analyze a large amount of data at the same time is not possible.

Introduction to SAP HANA

SAP HANA Goals

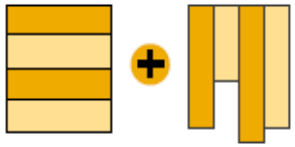
- Enable new applications and optimize existing applications
- High performance and scalability
- Hybrid data management system – combining different paradigms in one system
- Compatibility and standard Database Management System (DBMS) features
- Support for text analysis, indexing, and search
- Cloud support and application isolation
- Support for temporal tables
- Executing application logic inside the data layer

Introduction to SAP HANA

ABAP Meets SAP HANA: Evolution and Architecture

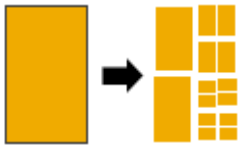
SAP Software Technology Innovations

Row and Column Store

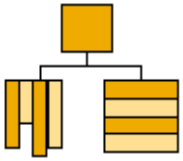


Data Compression

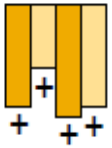
Table Partitioning



Avoidance of Aggregate Tables



Insert Only on Delta



Introduction to SAP HANA

Table partitioning:

- Table partitioning is a data organization scheme in which table data is divided across multiple storage objects called data partitions.

- In SAP HANA database, it is possible to split column-store tables horizontally into disjunctive sub-tables or partitions. The SAP HANA database supports several redistribution operations that use complex algorithms to evaluate the current distribution and determine a better distribution depending on the situation.

- Partitioning is typically used in distributed systems, but it may also be beneficial for single-host systems. Partitioning is transparent for SQL queries and data manipulation language statements.

Introduction to SAP HANA

- In a distributed SAP HANA system, tables are assigned to an index server on a particular host at their time of creation, but this assignment can be changed. In certain situations, it is even necessary.
- A non-partitioned table of SAP HANA database cannot store more than 2,000,000,000 (2 billion) rows.
- For example, in SAP HANA side-by-side implementation, SLT will stop data replication when SAP HANA table reaches 2 billion records.

Introduction to SAP HANA

Avoidance of aggregate table

- A very important part of the HANA philosophy is that all data should be stored at the highest possible level of granularity (e.g. the level of greatest detail). This is in contrast to the prevailing philosophy in most enterprise data centers, which says that the data should be stored on whatever level of granularity is required by the application to ensure maximum performance.

- Unfortunately, multiple applications use the same information and require different levels of detail, which results in high redundancy and software complexity around managing the consistency between multiple aggregate tables and source data.

- Given the incredible aggregation speed provided by HANA, all aggregates required by any application can now be computed from the source data on-the-fly, providing the same or better performance as before and dramatically decreasing code complexity which makes system maintenance a lot easier.

Introduction to SAP HANA

Insert only on Delta:

- Next innovation in SAP HANA is insert-only approach.

- This means that data in a table will not be overwritten when doing an update.
- Instead, a new entry with the same primary key at the end of the table is appended.

- The trick here is that HANA adds a transaction ID to the entry, and when selecting the data, only the entry with the highest transaction ID is returned.

Introduction to SAP HANA

- The deletion of a column will first of all only mark the entry as deleted and not remove it immediately from the table.

- Also, newly inserted data is not immediately added to the main table itself, but written in an uncompressed delta table which is transparent and automatically joined to the main table when selecting the data.

Evolution of ABAP for SAP HANA

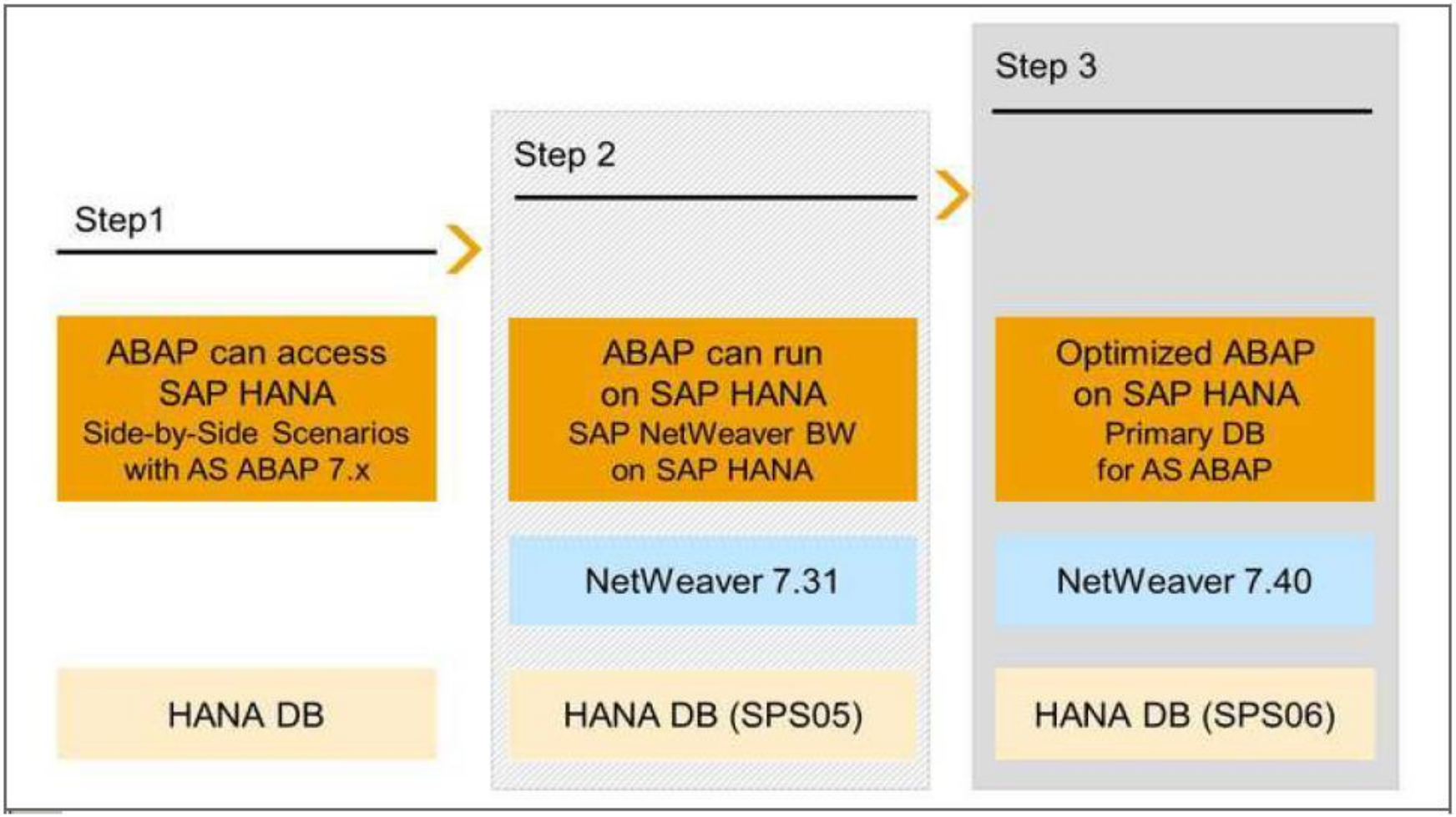
ABAP lays the foundation for countless SAP applications, with a broad range of features and functionality for creating powerful business solutions.

SAP HANA provides the opportunity to innovate with new and sophisticated technologies.

With support package stack (SPS) 05 for SAP NetWeaver 7.4, SAP has brought ABAP and SAP HANA together with features that enable developers to leverage the best of both worlds, not only in new custom development projects, but also in their existing applications.

Evolution of ABAP for SAP HANA

ABAP Programming for SAP HANA: Evolution and Opportunities



Evolution of ABAP for SAP HANA

Delivered as a globally available service release in December 2013, SPS 05 offers many new artifacts that ABAP developers can use to push down code to SAP HANA and bring innovation to their applications at various levels of abstraction.

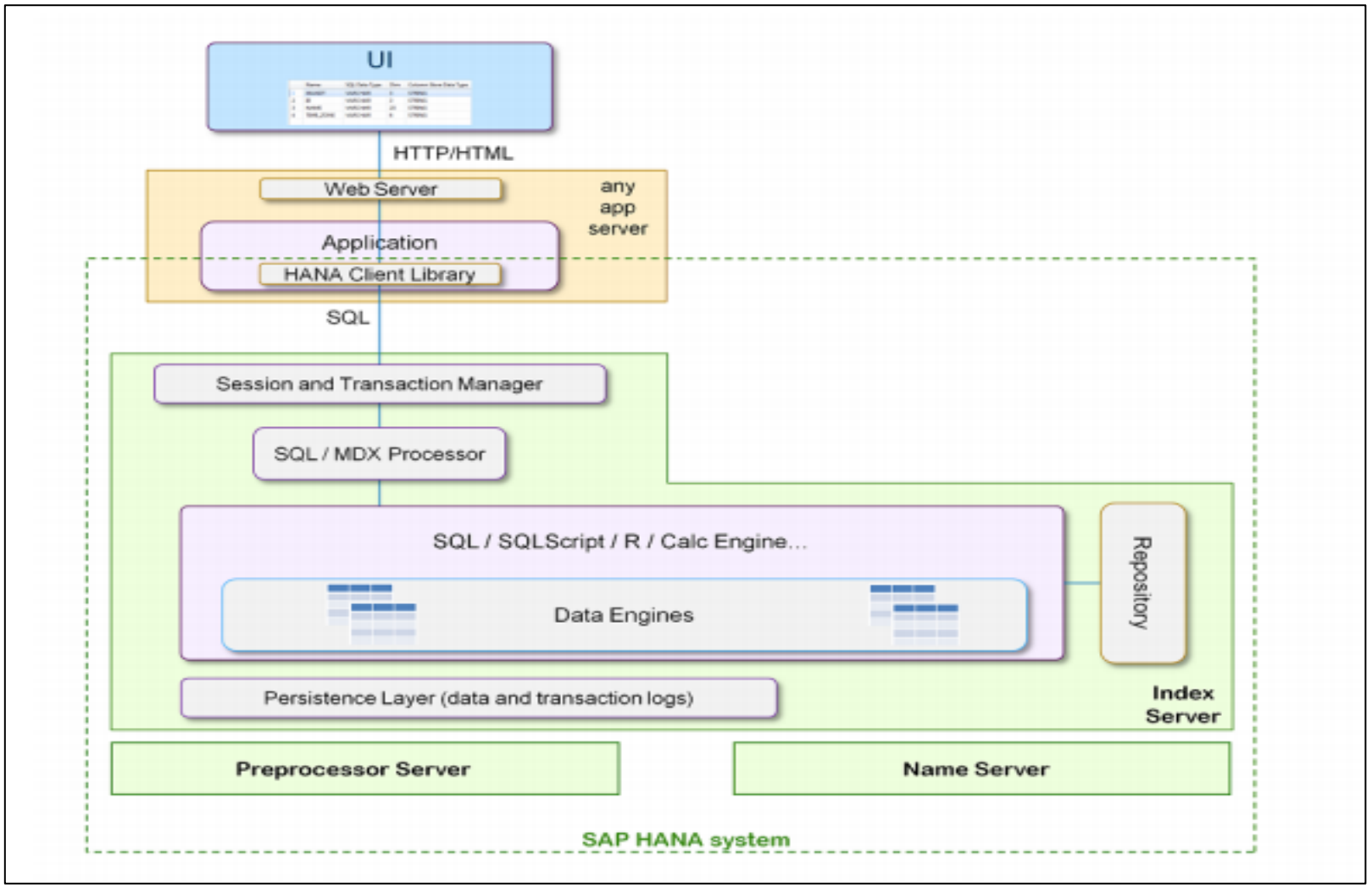
HANA Architecture

- The SAP HANA database is developed in C++ and runs on SUSE Linux Enterprise Server.

- SAP HANA database consists of multiple servers and the most important component is the Index Server.

- SAP HANA database consists of Index Server, Name Server, Statistics Server, Preprocessor Server and XS Engine.

HANA Architecture



HANA Architecture

Index Server:

- Index server is the main SAP HANA database component
- It contains the actual data stores and the engines for processing the data.
- The index server processes incoming SQL or MDX statements in the context of authenticated sessions and transactions.

Persistence Layer:

- The database persistence layer is responsible for durability and atomicity of transactions.
- It ensures that the database can be restored to the most recent committed state after a restart and that transactions are either completely executed or completely undone.

HANA Architecture

Preprocessor Server:

- The index server uses the preprocessor server for analyzing text data and extracting the information on which the text search capabilities are based.

Name Server :

- It owns the information about the topology of SAP HANA system.
- In a distributed system, the name server knows where the components are running and which data is located on which server.

Statistic Server :

- The statistics server collects information about status, performance and resource consumption from the other servers in the system.
- The statistics server also provides a history of measurement data for further analysis.

HANA Architecture

Session and Transaction Manager:

- The Transaction manager coordinates database transactions, and keeps track of running and closed transactions.
- When a transaction is committed or rolled back, the transaction manager informs the involved storage engines about this event so they can execute necessary actions.

XS Engine:

- XS Engine is an optional component.
- Using XS Engine clients can connect to SAP HANA database to fetch data via HTTP.

SAP In-Memory

SAP HANA is an in-memory database:

- It is a combination of hardware and software made to process massive real time data using In-Memory computing.
- It combines row-based, column-based database technology.
- Data now resides in main-memory (RAM) and no longer on a hard disk.
- It’s best suited for performing real-time analytics, and developing and deploying real-time applications.

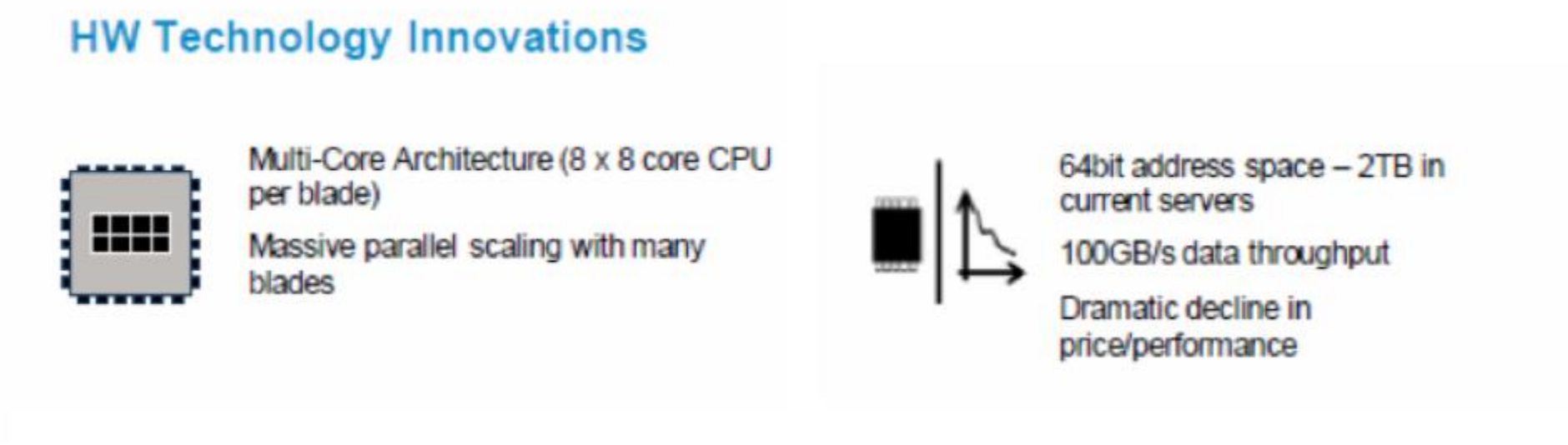
An in-memory database means all the data is stored in the memory (RAM).

The speed advantages offered by this RAM storage system are further accelerated by the use of multi-core CPUs, and multiple CPUs per board, and multiple boards per server appliance.

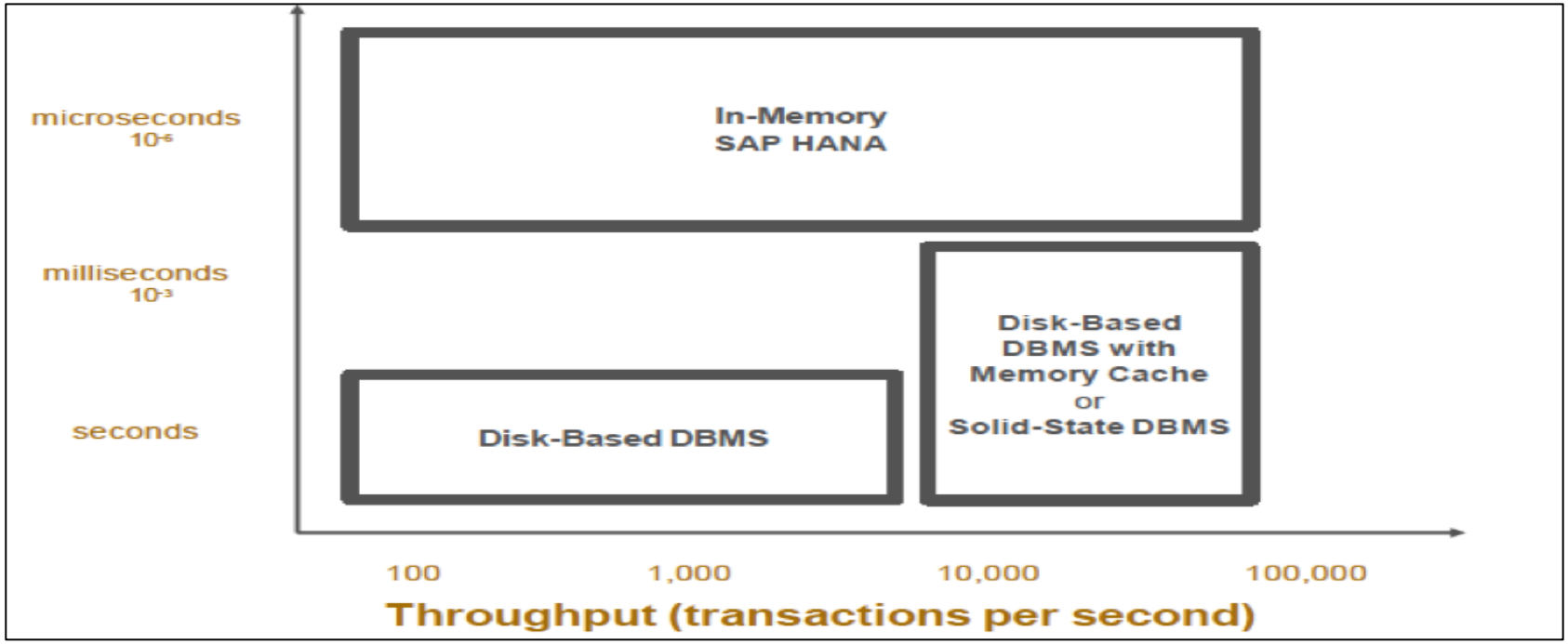
SAP In-Memory

SAP HANA is equipped with multiengine query processing environment which supports relational as well as graphical and text data within same system.

It provides features that support significant processing speed, handle huge data sizes and text mining capabilities.



SAP In-Memory

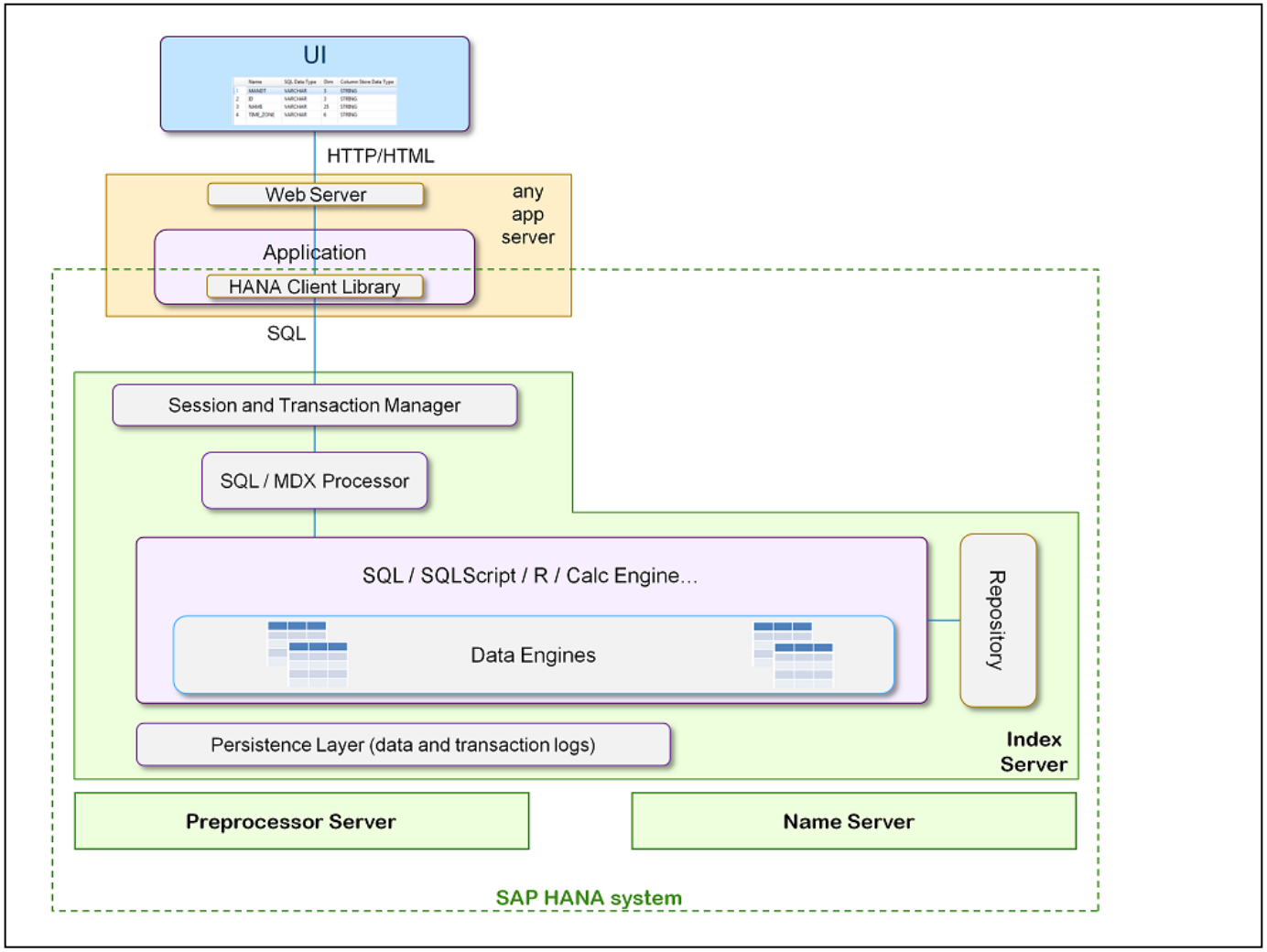


SAP HANA Database Concepts

A running SAP HANA system consists of multiple communicating processes (services).

The following shows the main SAP HANA database services in a classical application context.

SAP HANA Database Concepts



SAP HANA Database Concepts

- The SAP HANA database has its own scripting language named SQL Script.
- SQL Script embeds data-intensive application logic into the database.
- In addition to SQL Script, SAP HANA supports a framework for the installation of specialized and optimized functional libraries, which are tightly integrated with different data engines of the index server.
- SAP HANA also supports the development of programs written in the R language.

SAP HANA Database Concepts

- The main SAP HANA database management component is known as the index server, which contains the actual data stores and the engines for processing the data.

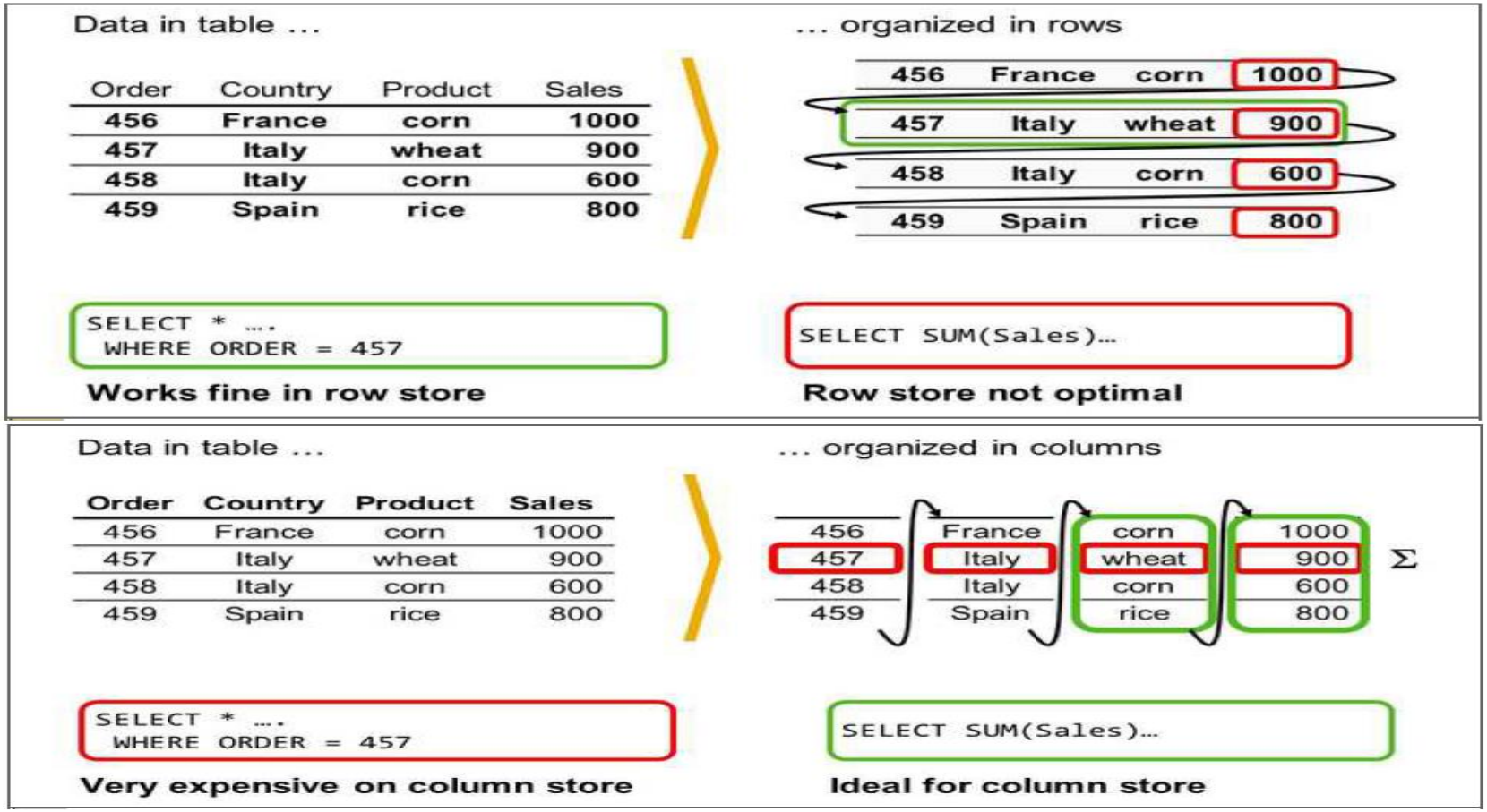
- The index server processes incoming SQL or MDX statements in the context of authenticated sessions and transactions.

SAP HANA Database Concepts

Row and Column Store:

- Row Storage - It stores table records in a sequence of rows.
- Column Storage - It stores table records in a sequence of columns i.e. the entries of a column is stored in contiguous memory locations.

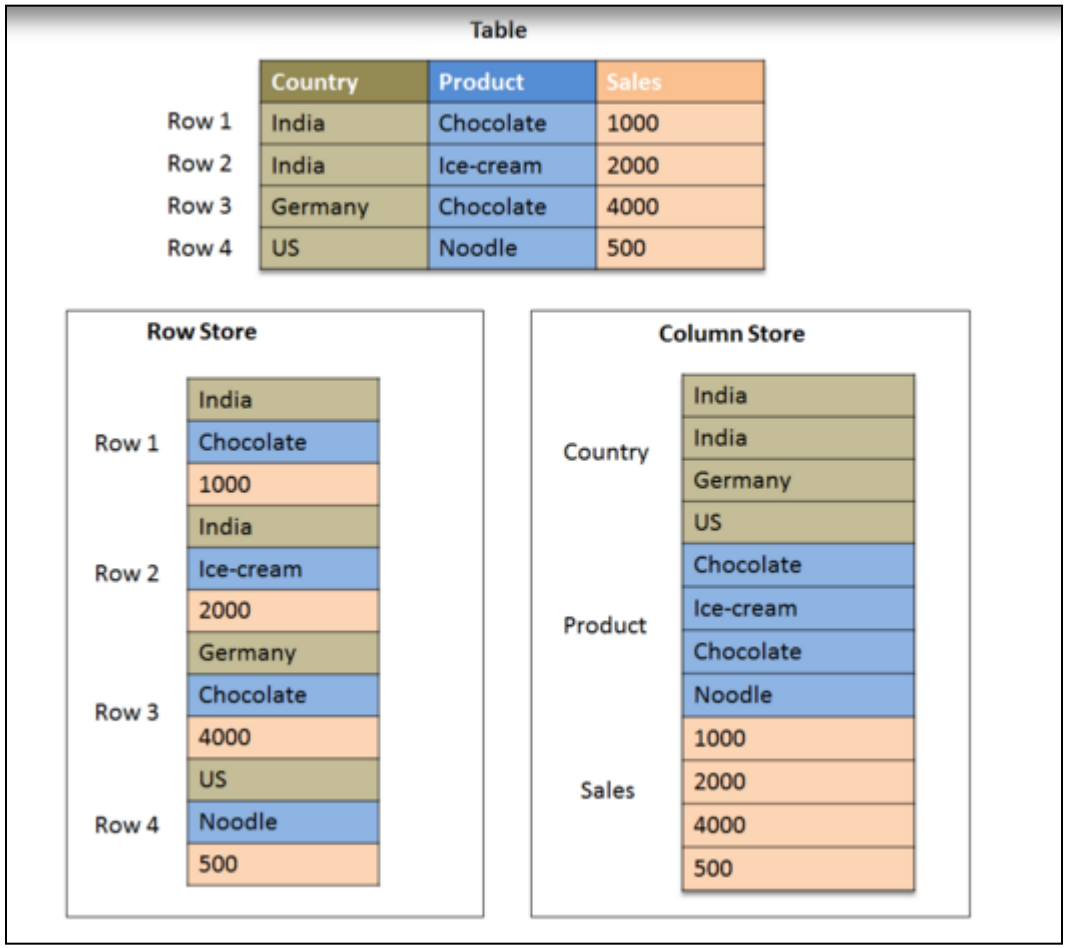
Column Store Versus Row Store



SAP HANA Database Concepts

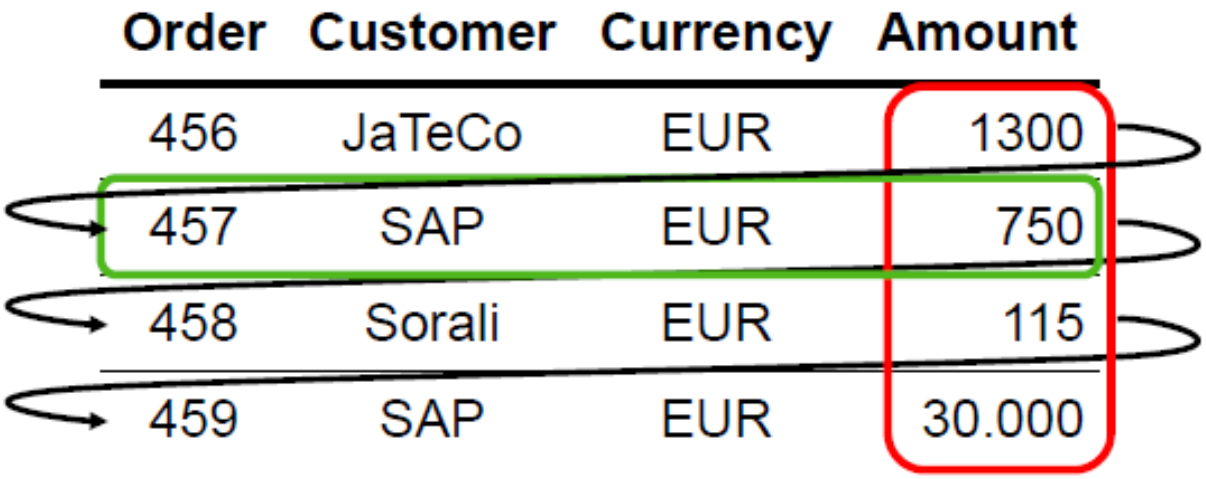
- SAP HANA Modeler Views can only be created on the top of Column based tables.
- Storing data in Column tables is not a new thing.
- Earlier it was assumed that storing data in Columnar based structure takes more memory size and not performance Optimized.
- With evolution of SAP HANA, HANA used column based data storage in Information views and presented the real benefits of columnar tables over Row based tables.

SAP HANA Database Concepts



SAP HANA Database Concepts

Row Store :



Order	Customer	Currency	Amount
456	JaTeCo	EUR	1300
457	SAP	EUR	750
458	Sorali	EUR	115
459	SAP	EUR	30.000

```
SELECT * ….
 WHERE ORDER = 457
```

Good performance

```
SELECT SUM(Amount)…
```

Low performance

SAP HANA Database Concepts

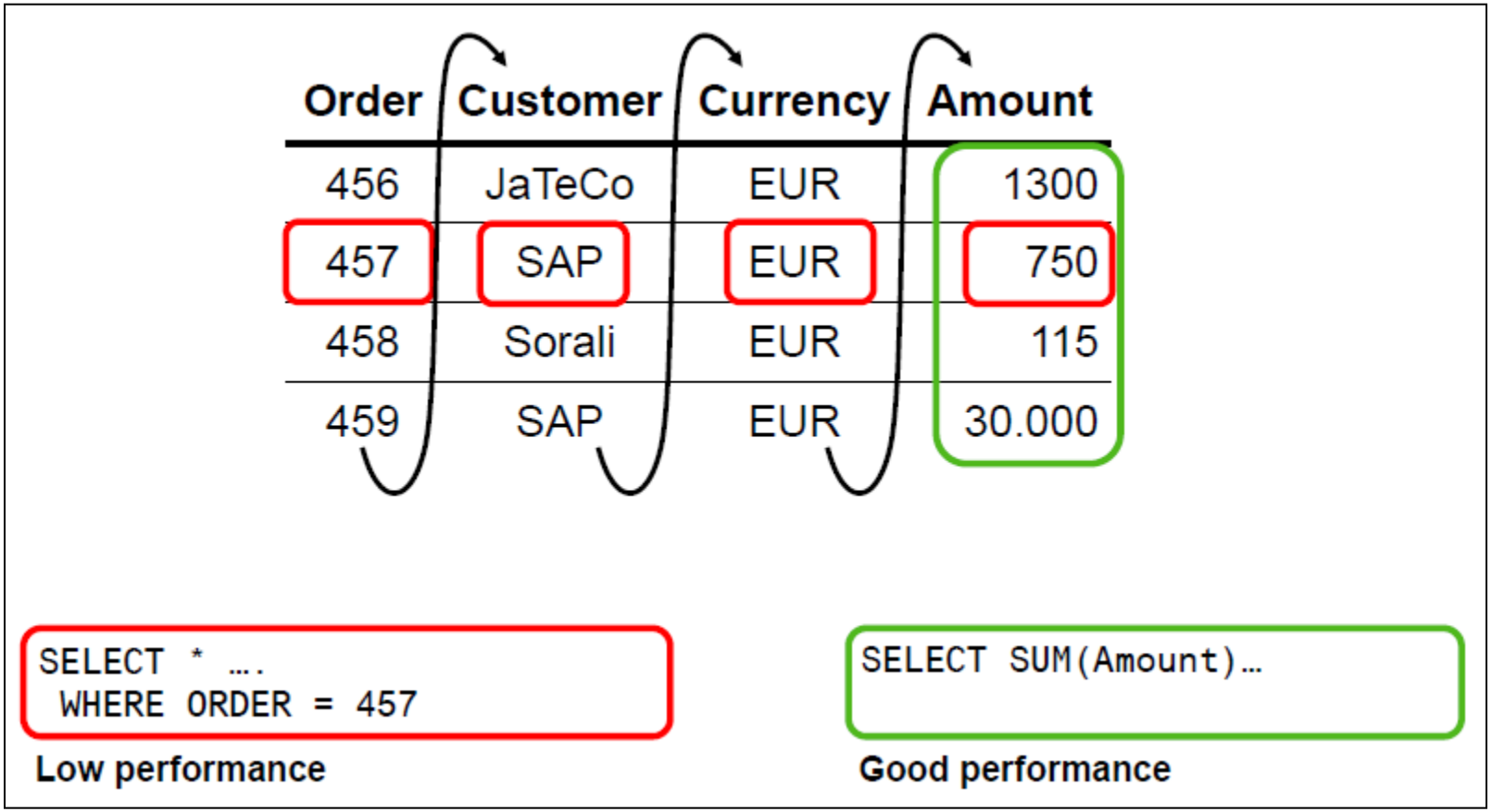
Column Store :

- In a conventional database, data is stored in Row based structure i.e. horizontally. SAP HANA stores data in both row and Column based structure. This provides Performance optimization, flexibility and data compression in HANA database.

- Storing Data in Columnar based table has following benefits:
  - Data Compression
  - Faster read and write access to tables as compared to conventional Row based storage
  - Flexibility & parallel processing
  - Perform Aggregations and Calculations at higher speed

SAP HANA Database Concepts

Column Store：



SAP HANA Database Concepts

- It is always advisable to use Column based storage, if SQL statement has to perform aggregate functions and calculations. Column based tables always perform better when running aggregate functions .

- Row based storage is preferred when output has to return complete row.

- The example given below makes it easy to understand.
  - While running a simple Select query, full row has to be printed in output so it is advisable to store table as Row based in this scenario.

SAP HANA Database Concepts

- Traditional databases store data simply in rows.
- The HANA in-memory database stores data in both rows and columns.
- It is this combination of both storage approaches that produces the speed, flexibility and performance of the HANA database.

SAP HANA Database Concepts

Advantages of column-based tables:

- Faster Data Access
- Better Compression
- Better parallel Processing

Advantages of Row Store :

- The application typically needs to access a complete record (or row).
- Neither aggregations nor fast searching are required.
- The table has a small number of rows (e. g. configuration tables, system tables).

SAP HANA Database Concepts

Disadvantages of Row Store:

- In case of analytic applications where aggregation are used and fast search and processing is required.
- In row based tables all data in a row has to be read even though the requirement may be to access data from a few columns.

SAP HANA Database Concepts

Difference between Row and Column Store:

Property	Row Store	Column Store	Reason
Memory Usage	Higher	Lower	Compression
Transactions	Faster	Slower	Modifications require updates to multiple columnar
Analytics	Slower even if indexed	Faster	Smaller dataset to scan, inherent indexing

Database Compression :

The column store allows for the efficient compression of data.

This makes it less costly for the SAP HANA database to keep data in main memory.

It also speeds up searches and calculations.

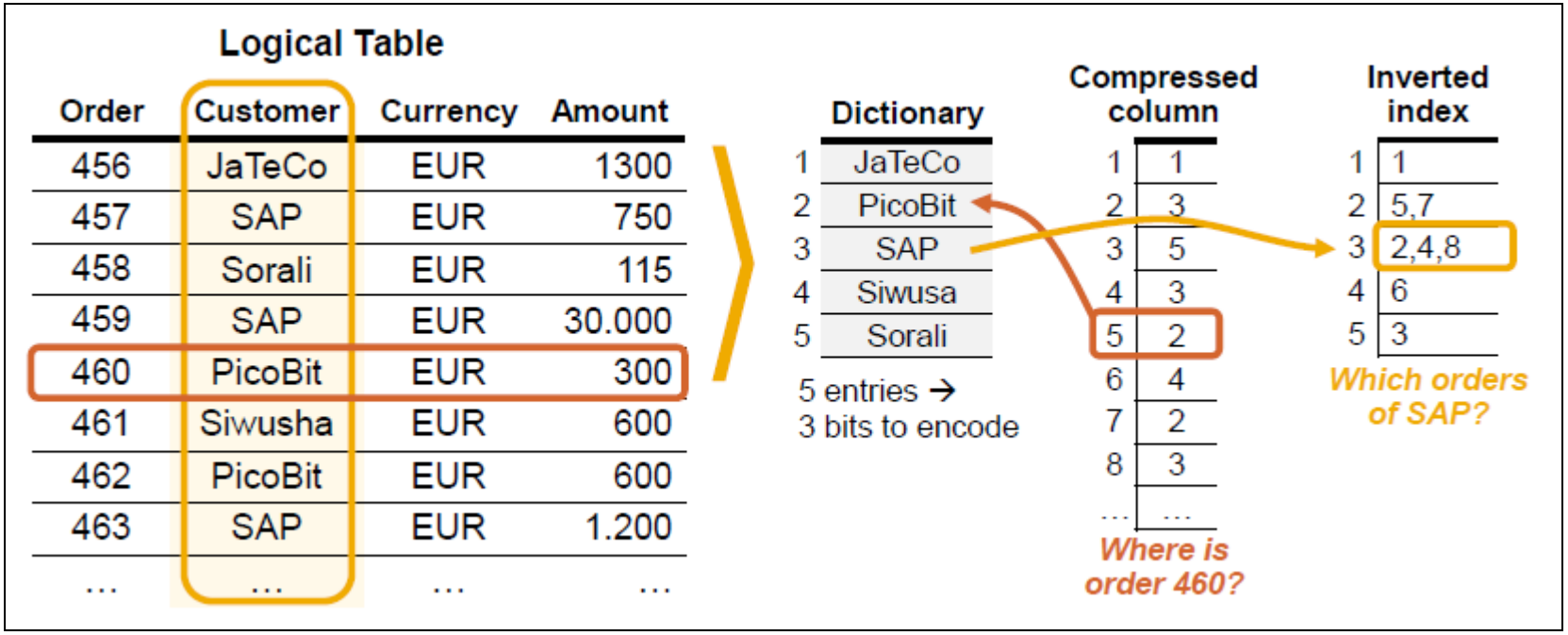
Data in column tables can have a two-fold compression:

- Dictionary compression
  - This default method of compression is applied to all columns. It involves the mapping of distinct column values to consecutive numbers, so that instead of the actual value being stored, the typically much smaller consecutive number is stored.
- Advanced compression
  - Each column can be further compressed using different compression methods, namely prefix encoding, run length encoding (RLE), cluster encoding, sparse encoding, and indirect encoding. The SAP HANA database uses compression algorithms to determine which type of compression is most appropriate for a column. Columns with the PAGE LOADABLE attribute are compressed with the NBit algorithm only.

Database Compression :

- Compression is automatically calculated and optimized as part of the delta merge operation.
- If you create an empty column table, no compression is applied initially as the database cannot know which method is most appropriate.
- As you start to insert data into the table and the delta merge operation starts being executed at regular intervals, data compression is automatically (re)evaluated and optimized.

Database Compression :



Database Compression :

Automatic compression optimization is ensured by the parameter active in the optimize compression section of the indexserver.ini configuration file. This parameter must have the value yes.

To store the contents of a column, the SAP HANA database creates a minimum of two data structures:

- a dictionary vector
- an attribute vector

Database Compression :

Record	Last name	Location	Gender
...	...	...	...
3	Brown	Chicago	M
4	Brown	San Francisco	F
5	Doe	Dallas	M
6	Doe	San Francisco	F
7	Smith	Dallas	M
...	...	...	...

Dictionary Vector
Last Name

Last Name	Position
...	...
Brown	7
Doe	8
Smith	9
...	...

Dictionary Vector
Location

Location	Position
...	...
Chicago	5
Dallas	6
San Francisco	7
...	...

Dictionary Vector
Gender

Gender	Position
F	1
M	2

Main Memory

...	7	7	8	8	9	...	5	7	6	7	6	...	2	1	2	1	2	...
...	3	4	5	6	7	...	3	4	5	6	7	...	3	4	5	6	7	...

Attribute Vector
Last Name

Attribute Vector
Location

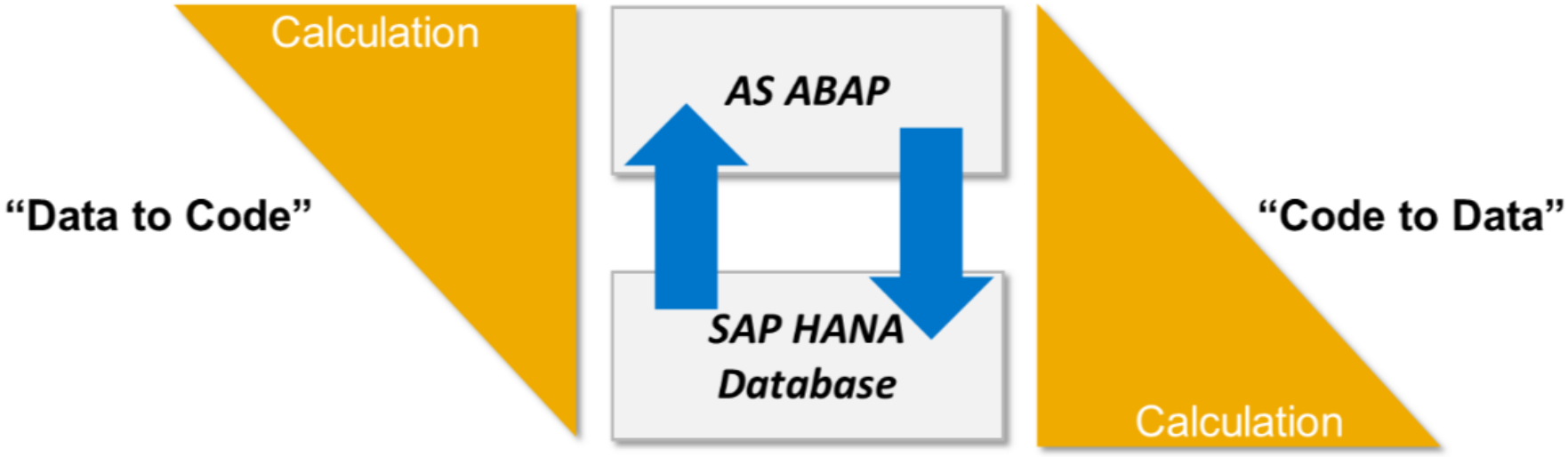
Attribute Vector
Gender

Code Pushdown:

One of the key differences for developing applications in ABAP for HANA is that you can push down data intense computations and calculations to the HANA DB layer instead bringing all the data to the ABAP layer and the processing the data to do computations.

This is what is termed as Code-to-Data paradigm in the context of developing ABAP applications optimized for HANA.

Code Pushdown:



Code Pushdown:

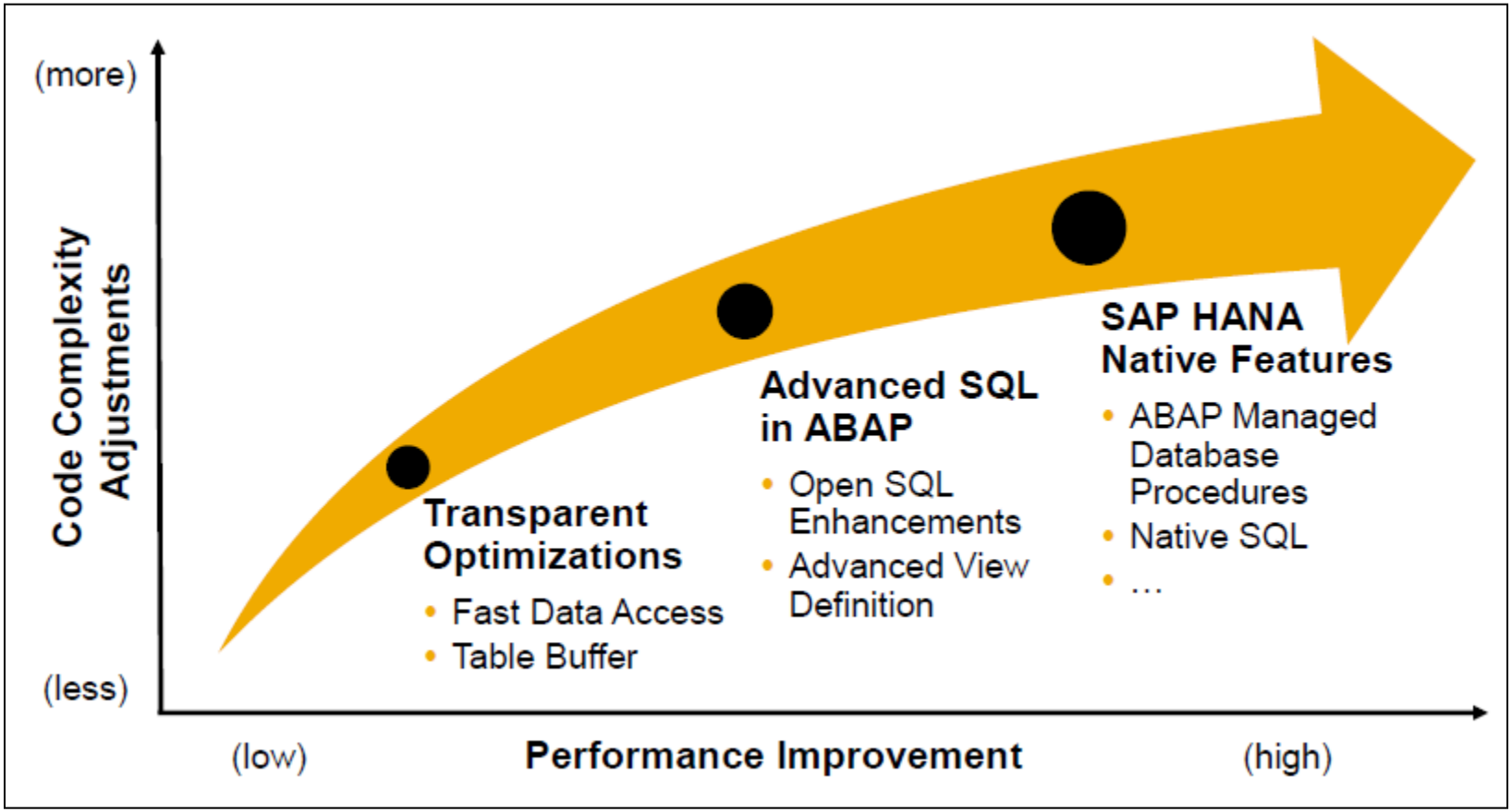
Code-to-Data Paradigm

- Code-to-data paradigm is basically a programming style in ABAP where you code to ‘push down’ data intensive computations and calculations to the HANA DB layer, instead of bringing all the data to the ABAP layer and then processing the data to do computations.

- When you plan to migrate to SAP HANA, the amount of custom code in your system will have an impact on total cost, time and the quality of the end result. In this article, we look at the potential impact of that custom code, and discuss how to tune your custom code to perform optimal on SAP HANA. By using the tips and tools mentioned here, you will be able to boost your ROI and smoothen the migration process.

Code Pushdown:

Code to Data – Performance Vs Code Adjustments :



Data Provisioning:

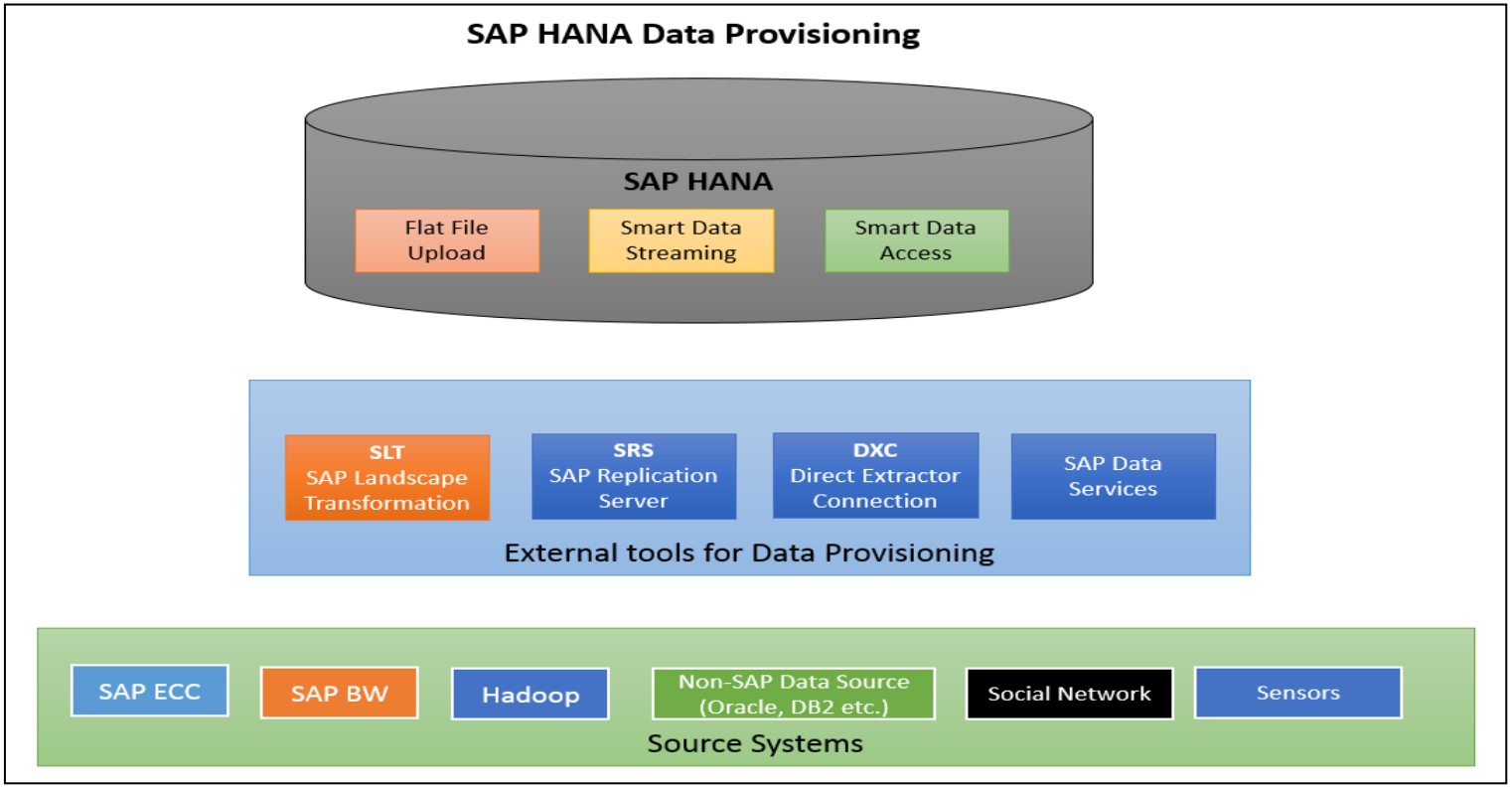
Data provisioning involves importing data from multiple SAP and non-SAP system into SAP HANA.

For HANA, we can divide the data provisioning options available into two categories, they are:

- SAP HANA in-built tools
- External Tools

Data Provisioning:

The pictorial representation of data provisioning techniques currently available to us for HANA.



Data Provisioning:

SAP Landscape Transformation Replication Server (“SLT”)

- is for all SAP HANA customers who need real-time or scheduled data replication, sourcing from SAP and NON-SAP sources
- Uses trigger-based technology to transfer the data from any source to SAP HANA in real-time.
- SLT server can be installed on the separate system or on SAP ECC System.

Data Provisioning:

SAP Replication Server (SRS)

- also known SAP Sybase Replication Server is a real-time data integration and movement software.

- SAP Replication Server (SRS) moves and synchronizes transactional data including DML and DDL across the enterprise, providing low impact, guaranteed data delivery, real-time business intelligence, and zero operational downtime.

Data Provisioning:

Direct Extractor Connection data replication

- reuses existing extraction, transformation, and load mechanism built into SAP Business Suite systems via a simple HTTP(S) connection to SAP HANA.
- It is a batch-driven data replication technique.
- It is considered as method for extraction, transformation, and load with limited capabilities for data extraction.

Data Provisioning:

SAP Data Services

- Gives a single enterprise level solution for data integration, transformation, data quality, data profiling and text data processing which allows us to:
  - Build a trusted data warehouse platform using data integration, data transform and data profiling
  - Provides single graphical user interface application for developers to build everything in the system.

DATA Provisioning Method Summary.

Method	Flat File Upload	SLT	ETL (BODS)	Sybase Replication	DXC
Use Case	Prototyping and Testing	Productive Usage	Productive Usage	Productive Usage	Productive Usage
Focus	Agility	Real-time	Extraction and Transformation	Real-time	Re-usability of SAP Content
Transformation Capabilities	None(1:1)	Default(1:1, conditional filtering and little transformation possible	Full range - support complex transformations	None(1:1)	Limited to function defined within function module of SAP Data Source

Summary

In this lesson, you have learnt:

- Basic Concepts Of SAP HANA
- Architecture Of SAP HANA
- In Memory Concept Used In SAP HANA
- Row And Column Storage In SAP HANA Database
- Data Provisioning.

Review Question

- SAP HANA Studio provides an environment for Administration, Modeling and Data Provisioning. (True/False)
- SAP HANA is a _____ database.
- Real time replication supported by _____.
- For small and extensive write operations,_____ store is advisable.
- SLT uses _____ technology to transfer data.

Review Question

- The SAP HANA database is developed in _____ and runs on _____ Server.
- SAP HANA database consists of _____ Server, _____ Server, _____ Server, _____ Server and _____ Engine.
- SAP HANA database cannot store more than _____rows for a non-partitioned table.