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 appliance is a hardware and software combination that integrates a number of SAP components
- SAP HANA is a flexible, data-source-agnostic appliance that enables customers to analyze large volumes of SAP-ERP data in real-time.

Introduction to SAP HANA

- SAP HANA Database is a hybrid in-memory database.
- SAP HANA combines row-based, column-based, and object-based database technology
- SAP HANA database consists of multiple servers
- SAP HANA database is developed in C++ and runs on SUSE Linux Enterprise Server

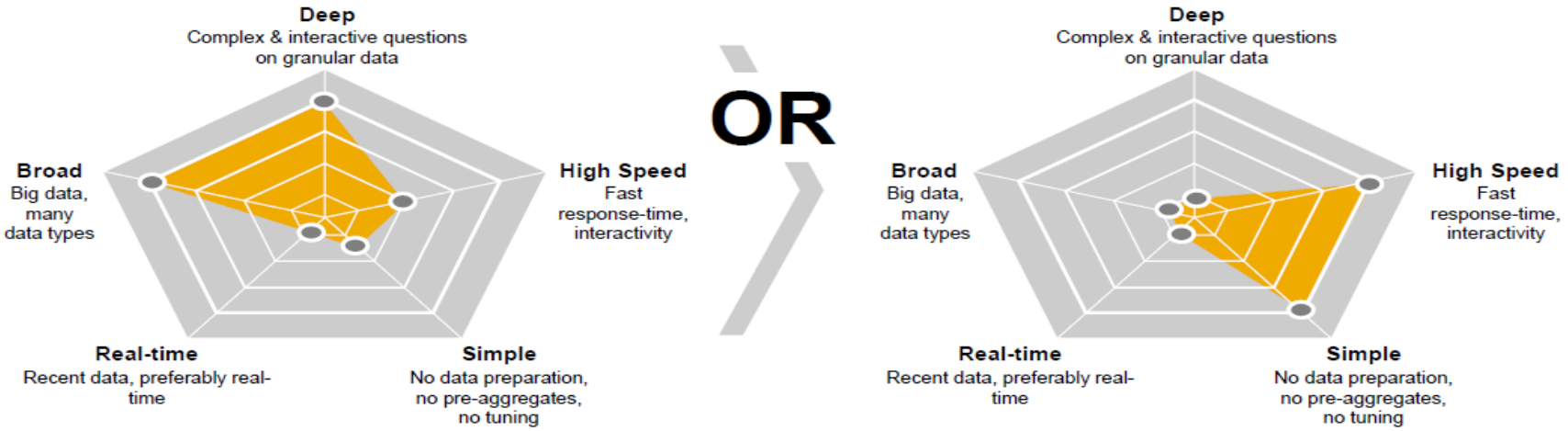
Introduction to SAP HANA

HANA is a 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 next 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

Traditional System involved losing one quality or aspect. something in return for gaining another quality or aspect

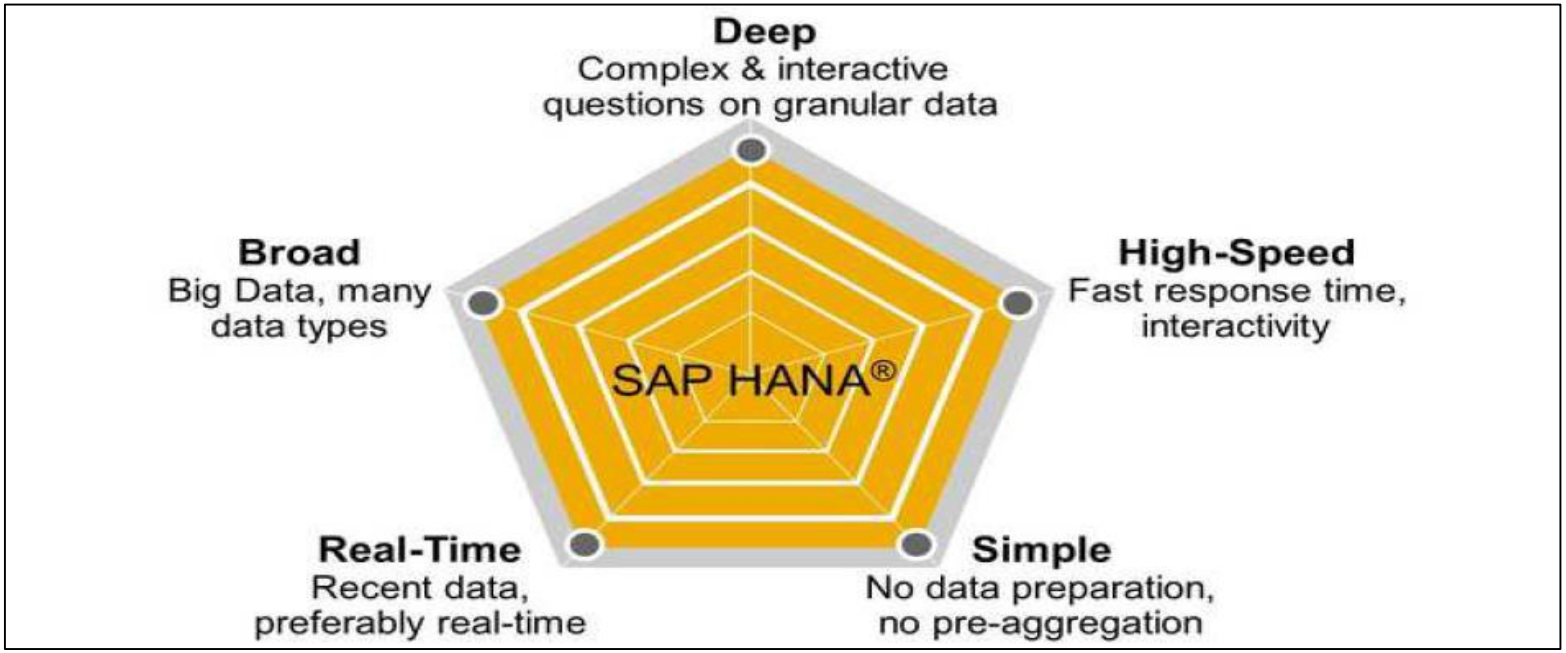


Before HANA, developing applications required several trade of decisions. Picture above, shows five typical dimensions while building a business applications.

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

HANA delivers all the five dimensions, so that you no longer have to make any 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.

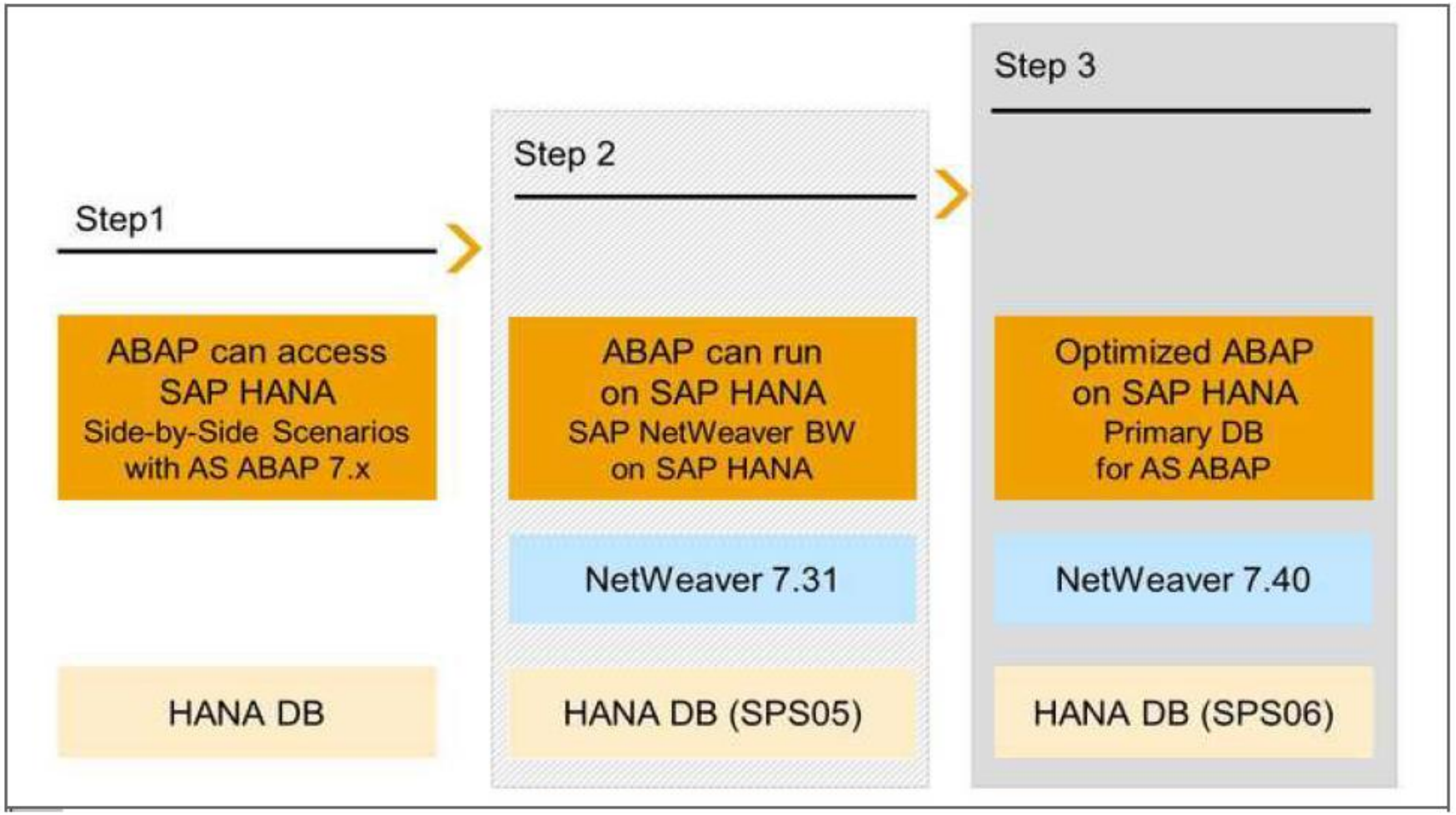
Evolution of ABAP for SAP HANA

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.

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.

Evolution of ABAP for SAP HANA



HANA Architecture

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

- SAP HANA database consists of

  Index Server,

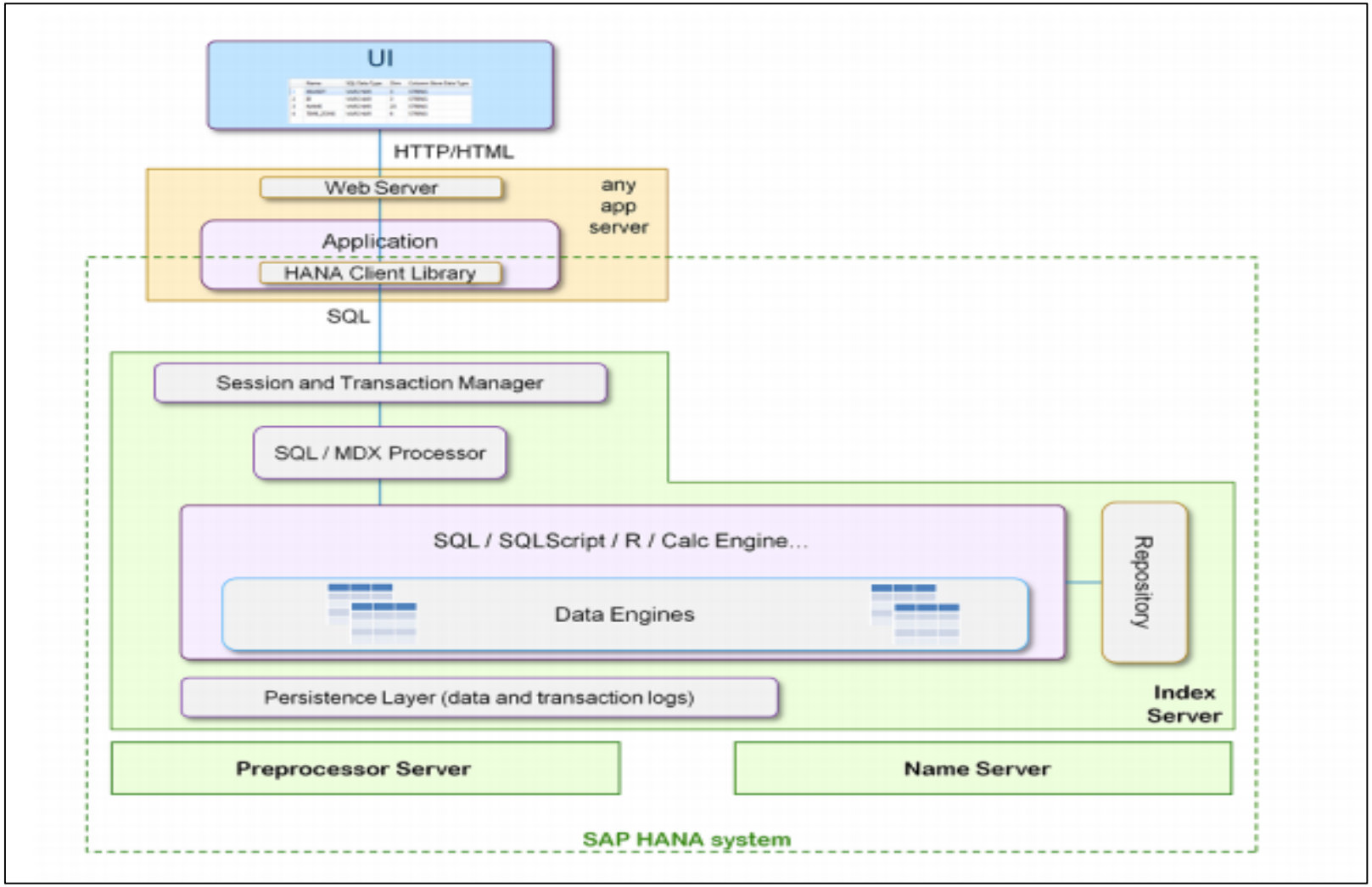
  Name Server,

  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:

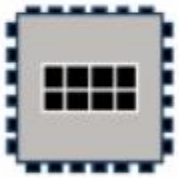
- An in-memory database means all the data is stored in the memory (RAM).
- Data now resides in main-memory (RAM) also.
- It’s best suited for performing real-time analytics, and developing and deploying real-time applications.
- The speed advantages offered by this RAM storage system are further accelerated by the use of multi-core CPUs, and multiple CPUs per board, and multiple boards per server appliance.

SAP In-Memory

It 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.

HW Technology Innovations



Multi-Core Architecture (8 x 8 core CPU per blade)

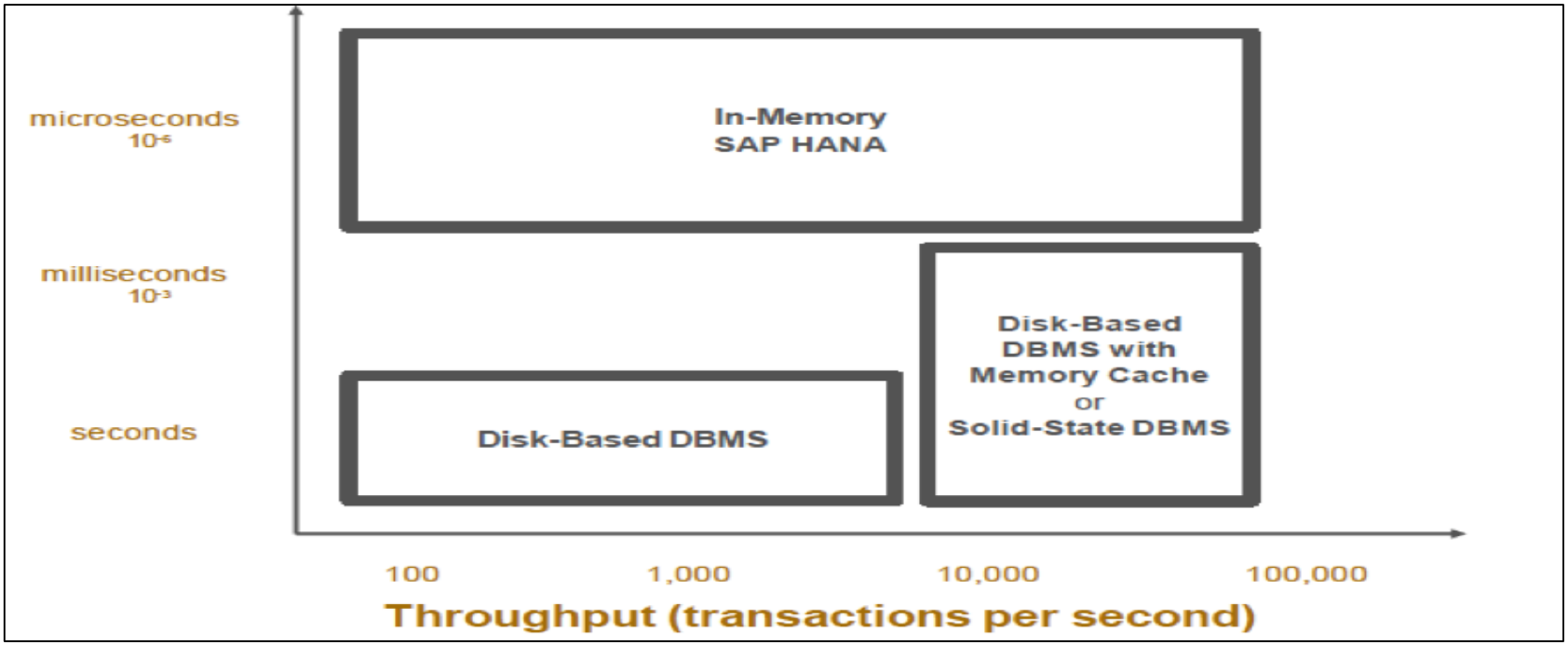
Massive parallel scaling with many blades

64bit address space – 2TB in current servers

100GB/s data throughput

Dramatic decline in price/performance

SAP In-Memory

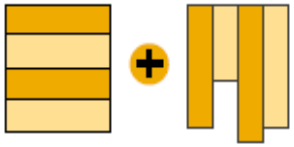


SAP HANA Database

- 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 Database

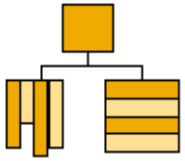
Row and Column Store



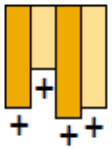
Data Compression

Table Partitioning

Avoidance of Aggregate Tables

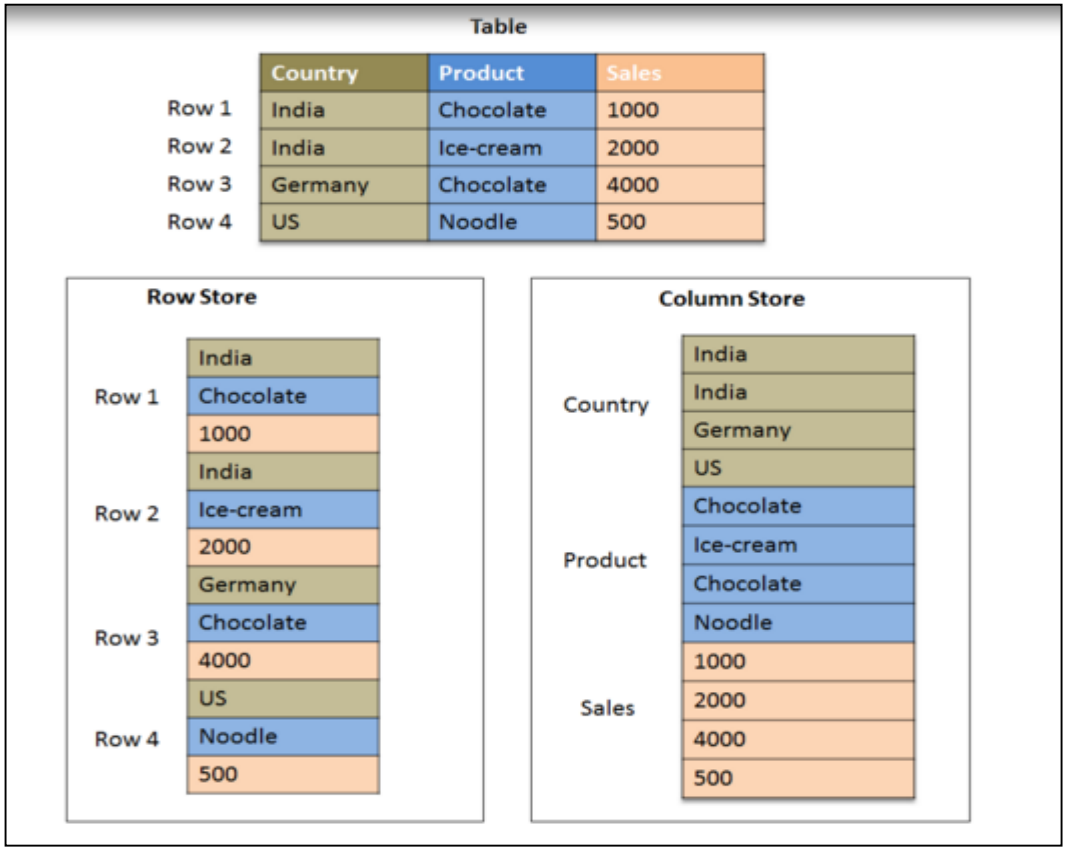


Insert Only on Delta

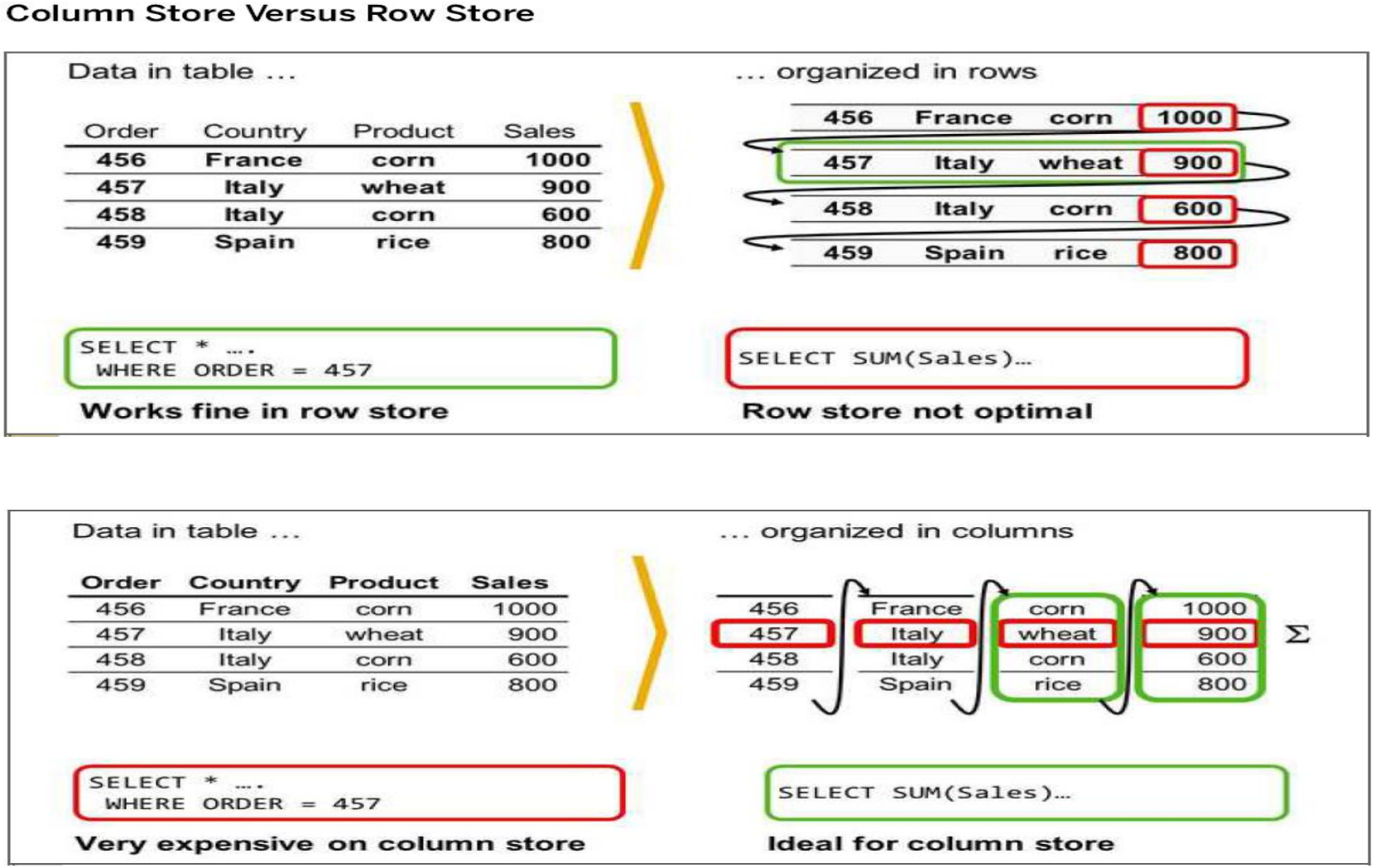


1. Row and Column Store

- 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.

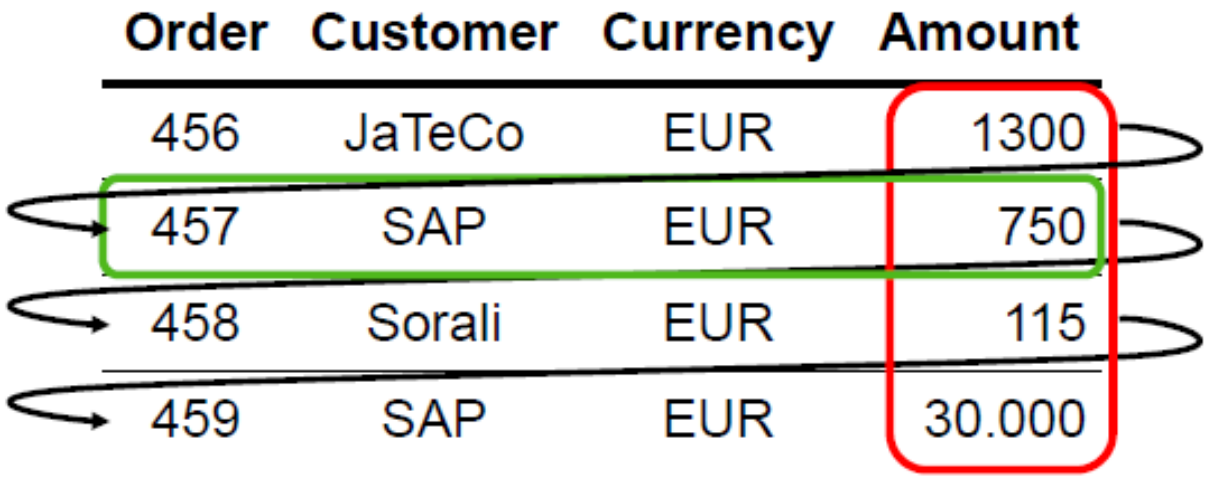


1. Row and Column Store



1. Row and Column Store

Row Store :




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

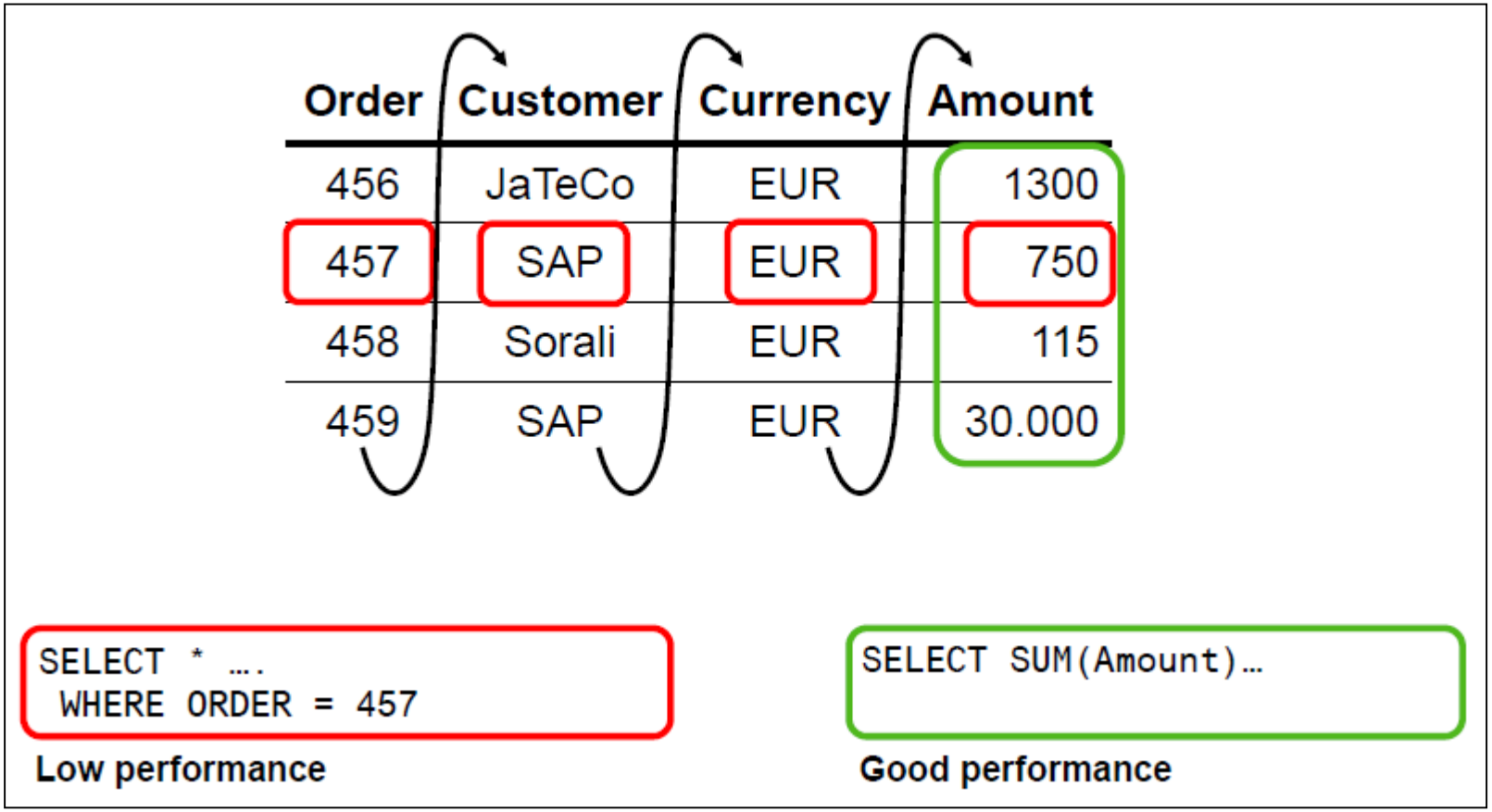
Good performance

```
SELECT SUM(Amount)…
```

Low performance

1. Row and Column Store

Column Store：



1. Row and Column Store

- In a conventional database, data is stored in Row based structure i.e. horizontally.
- 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.
- Storing data in Column tables is not a new thing.
- SAP HANA stores data in both row and Column based structure. This provides Performance optimization, flexibility and data compression in HANA database.

1. Row and Column Store

Row Store is used when :

- 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).
- For. Eg. 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

Disadvantage of Row Store:

- 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

1. Row and Column Store

Column Store is used when :

- If SQL statement has to perform aggregate functions and calculations

Advantages of column-based tables:

- Data Compression
- Faster read and write access to tables as compared to conventional Row based storage
- Flexibility & parallel processing
- SAP HANA Modeler Views can only be created on the top of Column based tables.
- It is always advisable to use Column based storage to perform calculations.

It is this combination of both storage approaches that produces the speed, flexibility and performance of the HANA database

1.Row and Column Store

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

2.Data Compression :

- Compression makes it less costly for the SAP HANA database to keep data in main memory.
- The column store allows for the efficient compression of data. It also speeds up searches and calculations.
- Compression is automatically calculated and optimized as part of the delta merge operation.
- The compressed table size is the size occupied by the table in the main memory of SAP HANA database

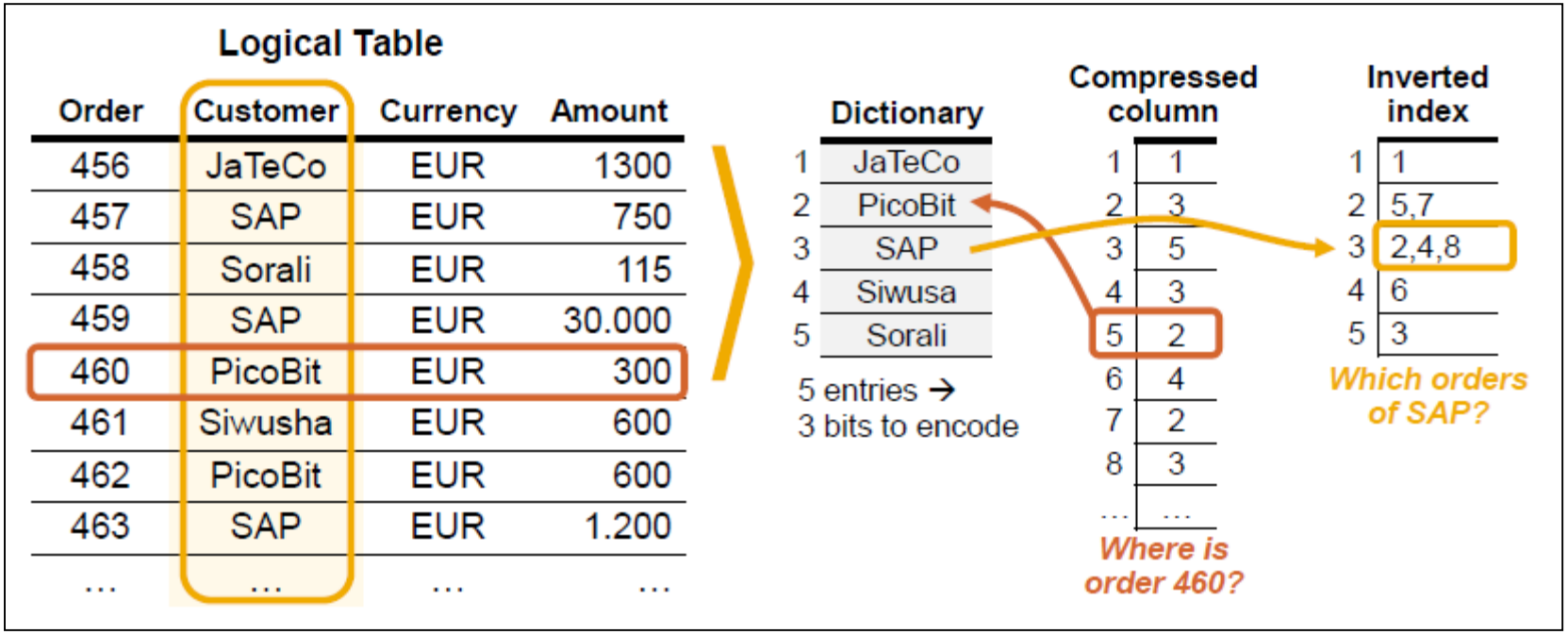
2. Data Compression :

In SAP HANA, there are various methods and algorithms that you can use to store data in Column based structure- Dictionary Compressed, attribute Vector Compressed and many more.

- **A dictionary vector:** Dictionary Compressed, 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

- **An attribute vector :** In attribute vector Compressed, cells are stored in form attribute vectors

2.Data Compression : Dictionary Vector



2.Data Compression : Attribute Vector

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

3. Table partitioning

Table partitioning:

- Table partitioning is a data organization scheme in which table data is divided across multiple storage objects called data partitions.
- The partitioning feature of the SAP HANA database splits column-store tables horizontally into disjunctive sub-tables or partitions. In this way, large tables can be broken down into smaller, more manageable parts.
- Partitioning is typically used in multiple-host systems, but it may also be beneficial in single-host systems..
- When a table is partitioned, the split is done in such a way that each partition contains a different set of rows of the table
- There are several alternatives available for specifying how the rows are assigned to the partitions of a table, for example, hash partitioning or partitioning by range.

3. Table partitioning

Following are the typical advantages of partitioning.

- **Load balancing in a distributed system**: Individual partitions can be distributed across multiple hosts. This means that a query on a table is not processed by a single server but by all the servers that host partitions

- **Overcoming the size limitation of column-store tables**: A non-partitioned table cannot store more than 2 billion rows. It is possible to overcome this limit by distributing the rows across several partitions. Each partition must not contain more than 2 billion rows.

- **Parallelization** Partitioning allows operations to be parallelized by using several execution threads for each table.

4. Avoidance of Aggregate Tables

Prevailing philosophy in most traditional enterprise data centers, 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.

This results in high redundancy and software complexity around managing the consistency between multiple aggregate tables and source data.

HANA philosophy is that all data should be stored at the highest possible level of granularity.

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.

5. Insert only on Delta

- In the column store HANA will only insert data. That is whenever you update or delete data, HANA will just insert.
- For performance reasons, the column store is divided in a main and a delta part. New data is going to the delta and will be merged frequently to the main
- Changing statements work as follows:

INSERT:

An INSERT statement will just *insert* a new record in the delta. The merge process will move the record from delta to main.

DELETE:

A DELETE statement will select the record and mark it as invalid by setting a flag (for main or delta). The merge process will delete the record from memory once there is no open transaction active for it anymore.

5. Insert only on Delta

UPDATE:

An UPDATE statement will insert a new version of the record. The merge process will move the latest version from delta to main. Old versions will be deleted once there is no open transaction active for them anymore. The update therefore consists internally of an insert + delete

And what are the benefits of the insert only approach?

Since the main store is compressed changes would be expensive. Just imagine you want to make a change in a compressed text file. You need to uncompress, make the change and compress again

For this reason, data is only appended to the uncompressed delta store where change operations are much cheaper. The merge moves the data from delta to main and thus compresses only from **time to time** and not with every change.

Code Pushdown:

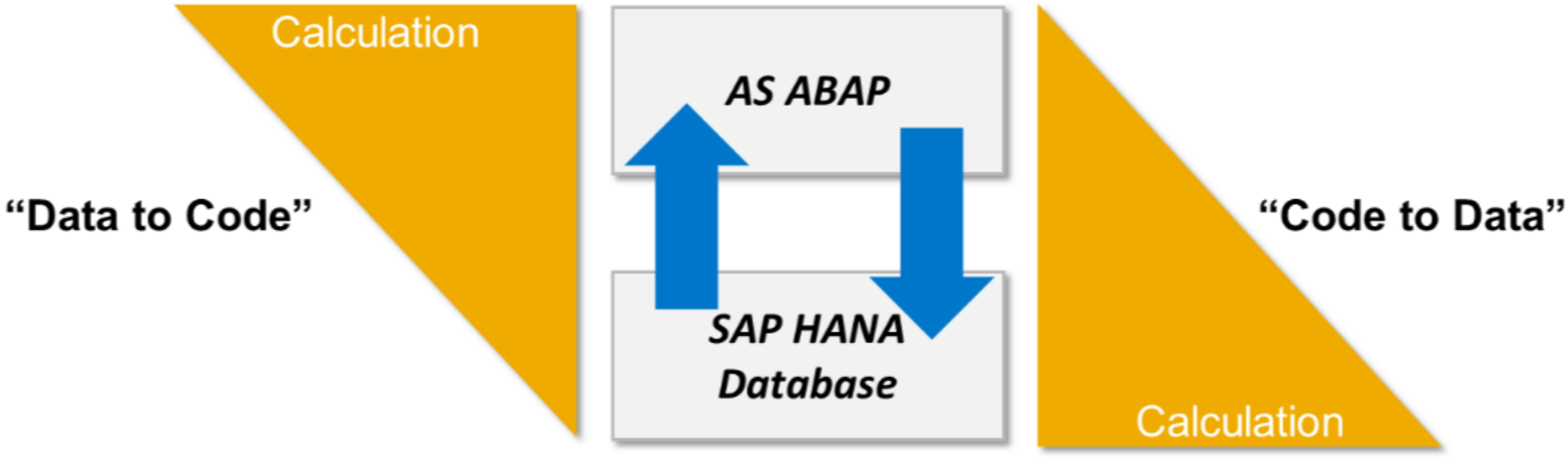
In HANA, you can push down data intense computations and calculations to the HANA DB layer instead of bringing all the data to the ABAP layer and then processing the data to do computations.

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

It is basically a programming style in ABAP where you code to ‘push down’ data intensive computations to the HANA DB layer.

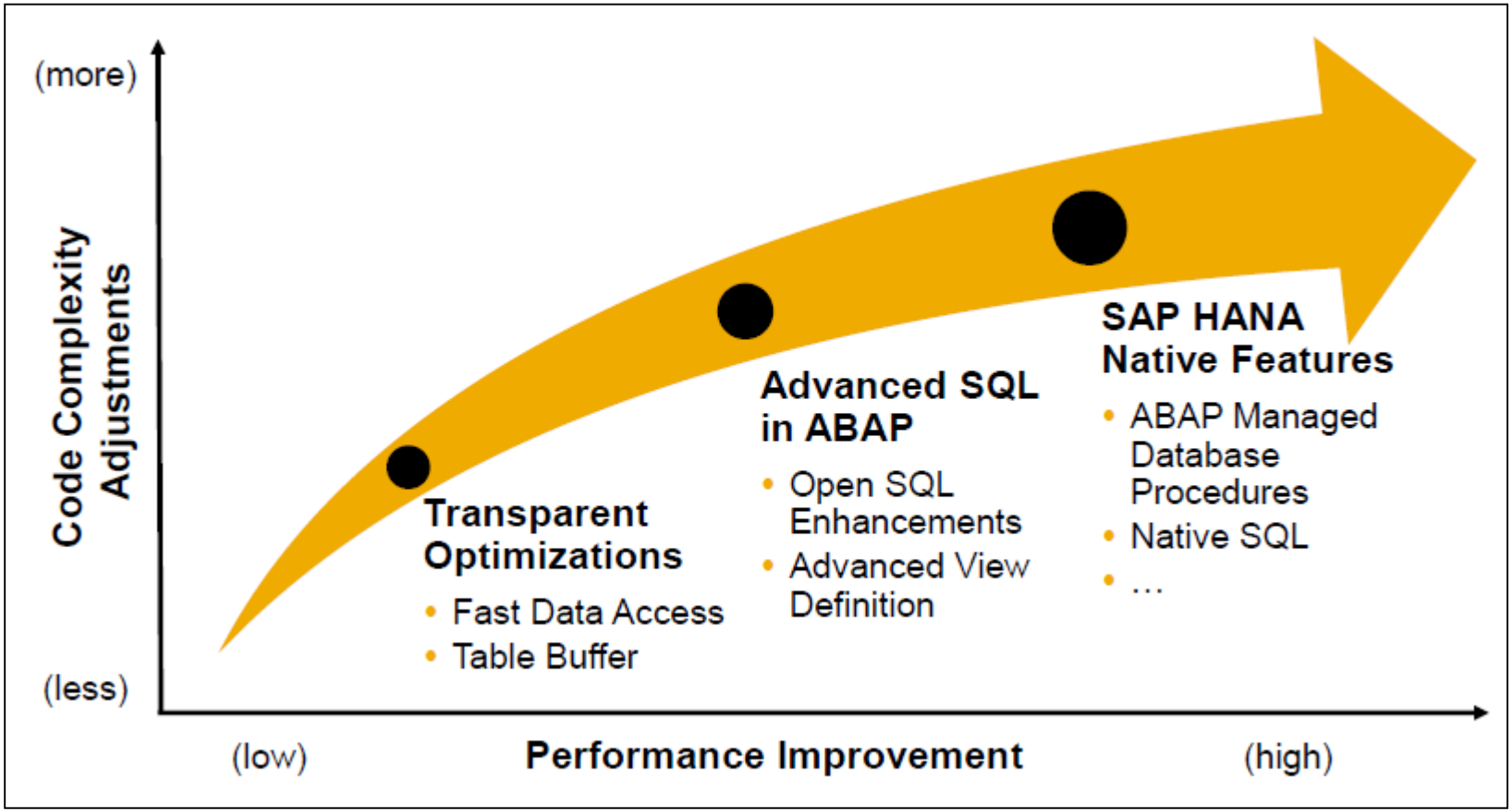
It is one of the key differences for developing applications in ABAP for HANA.

Code Pushdown:



Code Pushdown:

Code to Data – Performance Vs Code Adjustments :



Data Provisioning:

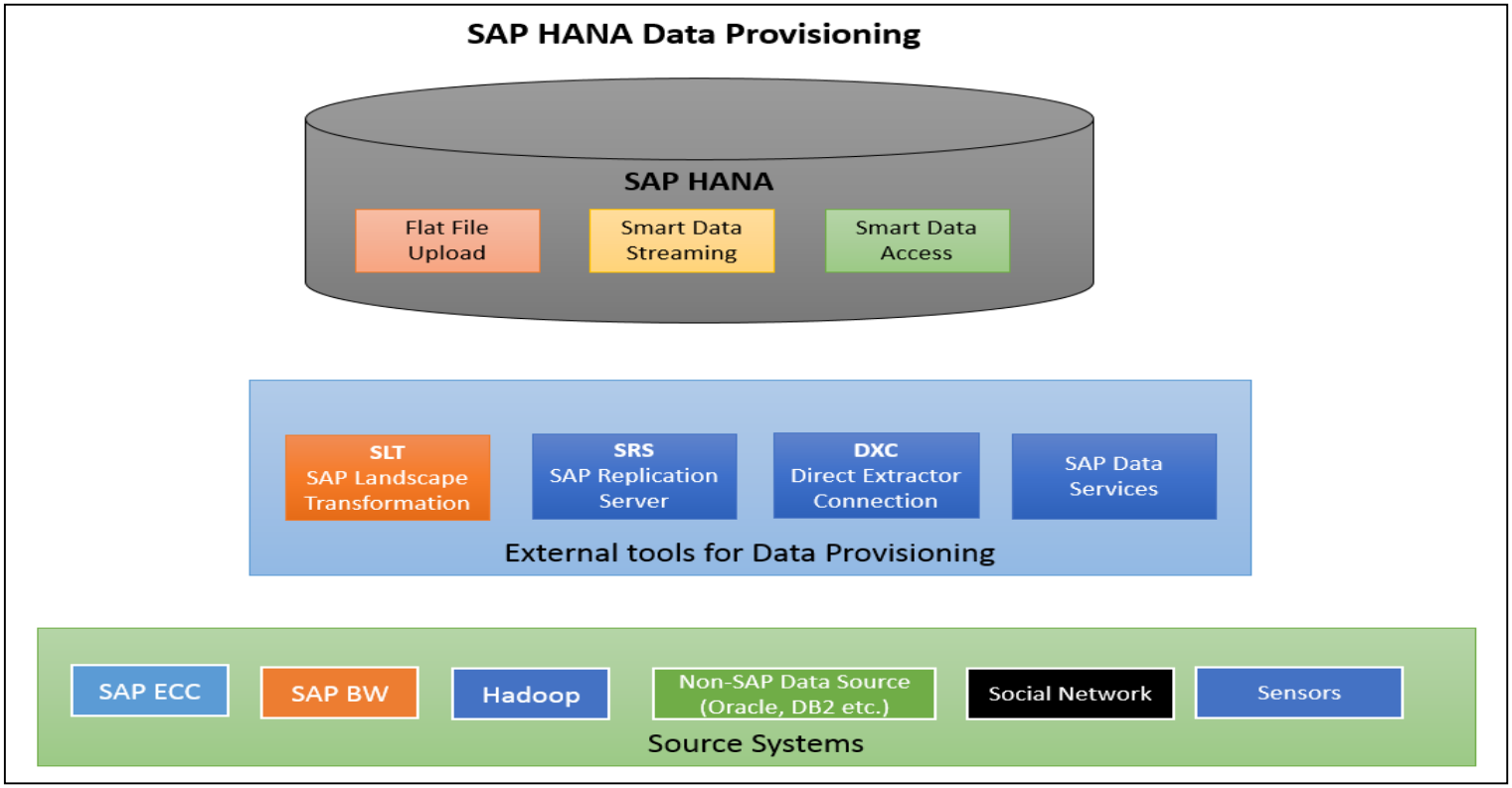
Data provisioning involves importing data from multiple SAP and non-SAP system into SAP HANA.

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.

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.