Data Partitioning and Partitioning strategies

Agenda

- Partition
- Why is it Necessary to Partition?
- Horizontal Partitioning
  - Partition on a Different Dimension
- Vertical Partitioning
  - Normalization
  - Row splitting
- Functional partitioning
- Tool Data Partitioning
- Technique of hive Partition

Partitioning

- Partitioning is done to enhance performance and facilitate easy management of data.
- Partitioning also helps in balancing the various requirements of the system.
- It optimizes the hardware performance and simplifies the management of data warehouse by partitioning each fact table into multiple separate partitions

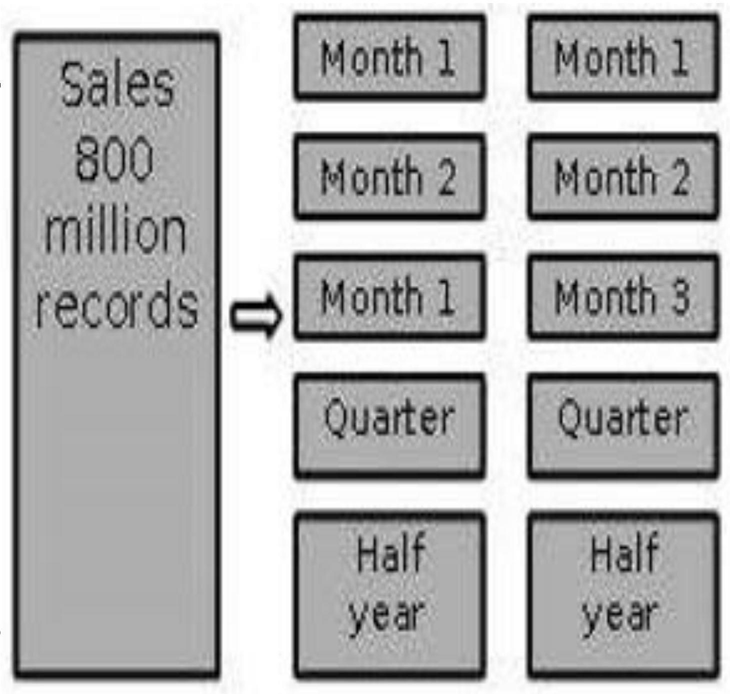
Why is it Necessary to Partition?

Partitioning is important for the following Reasons

- For easy management
- To assist backup/recovery
- To enhance performance
- Improve scalability
- Improve availability
- Improve security
- Provide operational flexibility
- Match the data store to the pattern of use

Horizontal Partitioning

- Horizontal partitioning divides a table into multiple tables. Each table then contains the same number of columns, but fewer rows. For example, a table that contains 1 billion rows could be partitioned horizontally into 12 tables, with each smaller table representing one month of data for a specific year. Any queries requiring data for a specific month only reference the appropriate table.
- Determining how to partition the tables horizontally depends on how data is analyzed. You should partition the tables so that queries reference as few tables as possible. Otherwise, excessive UNION queries, used to merge the tables logically at query time, can affect performance. For more information about querying horizontally partitioned tables



Horizontal Partitioning

- Partitioning data horizontally based on age and use is common. For example, a table may contain data for the last five years, but only data from the current year is regularly accessed. In this case, you may consider partitioning the data into five tables, with each table containing data from only one year.

Points to Note for Horizontal Partitioning

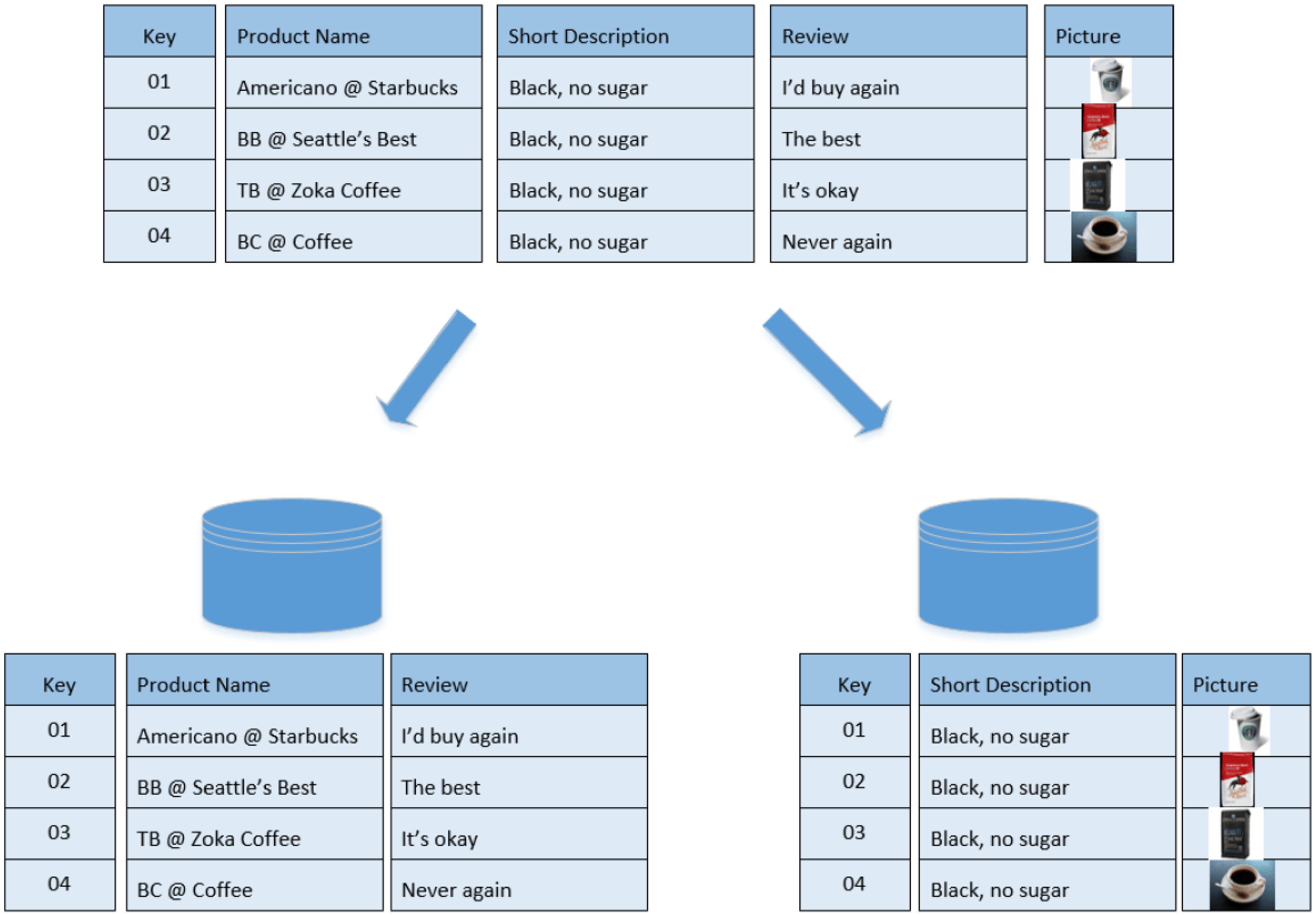
- The detailed information remains available online.
- The number of physical tables is kept relatively small, which reduces the operating cost.
- This technique is suitable where a mix of data dipping recent history and data mining through entire history is required.
- This technique is not useful where the partitioning profile changes on a regular basis, because repartitioning will increase the operation cost of data warehouse.

Partition on a Different Dimension (for Horizontal Partitioning)

- The fact table can also be partitioned on the basis of dimensions other than time such as product group, region, supplier, or any other dimension. Let's have an example.
- Suppose a market function has been structured into distinct regional departments like on a **state by state** basis. If each region wants to query on information captured within its region, it would prove to be more effective to partition the fact table into regional partitions. This will cause the queries to speed up because it does not require to scan information that is not relevant.

Vertical Partitioning

- Vertical partitioning divides a table into multiple tables that contain fewer columns. The two types of vertical partitioning are normalization and row splitting:
- <u>Normalization</u> is the standard database process of removing redundant columns from a table and putting them in secondary tables that are linked to the primary table by primary key and foreign key relationships.
- <u>Row splitting</u> divides the original table vertically into tables with fewer columns. Each logical row in a split table matches the same logical row in the other tables as identified by a UNIQUE KEY column that is identical in all of the partitioned tables. For example, joining the row with ID 712 from each split table re-creates the original row.



Key	Product Name	Short Description	Review	Picture
01	Americano @ Starbucks	Black, no sugar	I'd buy again	
02	BB @ Seattle's Best	Black, no sugar	The best	
03	TB @ Zoka Coffee	Black, no sugar	It's okay	
04	BC @ Coffee	Black, no sugar	Never again	

Key	Product Name	Review
01	Americano @ Starbucks	I'd buy again
02	BB @ Seattle's Best	The best
03	TB @ Zoka Coffee	It's okay
04	BC @ Coffee	Never again

Key	Short Description	Picture
01	Black, no sugar	
02	Black, no sugar	
03	Black, no sugar	
04	Black, no sugar	

Row Splitting

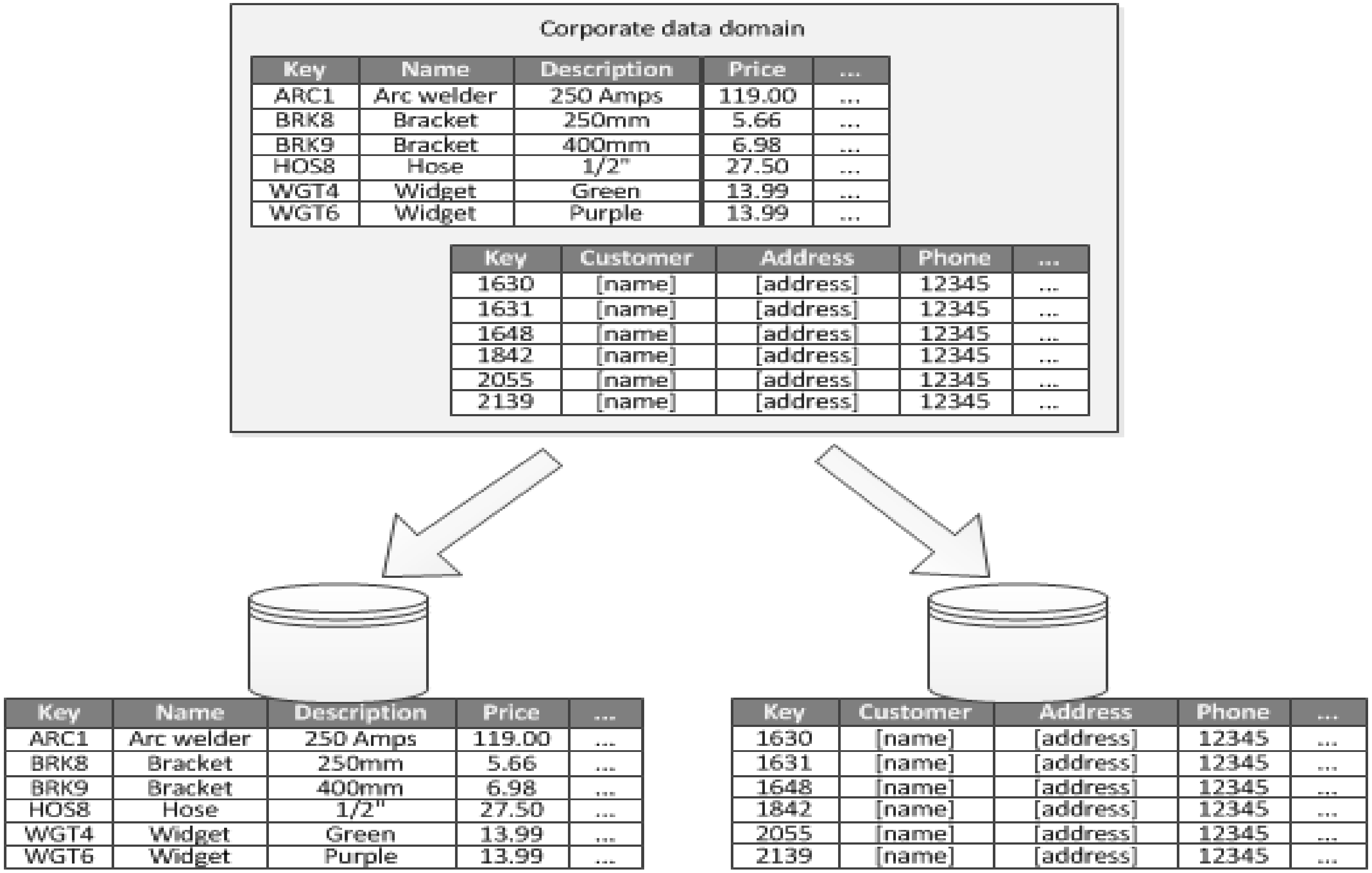
- Row splitting tends to leave a one-to-one map between partitions. The motive of row splitting is to speed up the access to large table by reducing its size.
- **Note** – While using vertical partitioning, make sure that there is no requirement to perform a major join operation between two partitions.

1. Like horizontal partitioning, vertical partitioning lets queries scan less data. This increases query performance. For example, a table that contains seven columns of which only the first four are generally referenced may benefit from splitting the last three columns into a separate table.
2. Vertical partitioning should be considered carefully, because analyzing data from multiple partitions requires queries that join the tables. Vertical partitioning also could affect performance if partitions are very large.

Functional partitioning

- For systems where it is possible to identify a bounded context for each distinct business area or service in the application, functional partitioning provides a technique for improving isolation and data access performance. Another common use of functional partitioning is to separate read-write data from read-only data that's used for reporting purposes. Figure 3 shows an overview of functional partitioning where inventory data is separated from customer data.

Functional Partitioning



Key	Name	Description	Price	...
ARC1	Arc welder	250 Amps	119.00	...
BRK8	Bracket	250mm	5.66	...
BRK9	Bracket	400mm	6.98	...
HOS8	Hose	1/2"	27.50	...
WGT4	Widget	Green	13.99	...
WGT6	Widget	Purple	13.99	...

Key	Customer	Address	Phone	...
1630	[name]	[address]	12345	...
1631	[name]	[address]	12345	...
1648	[name]	[address]	12345	...
1842	[name]	[address]	12345	...
2055	[name]	[address]	12345	...
2139	[name]	[address]	12345	...

Key	Name	Description	Price	...
ARC1	Arc welder	250 Amps	119.00	...
BRK8	Bracket	250mm	5.66	...
BRK9	Bracket	400mm	6.98	...
HOS8	Hose	1/2"	27.50	...
WGT4	Widget	Green	13.99	...
WGT6	Widget	Purple	13.99	...

Key	Customer	Address	Phone	...
1630	[name]	[address]	12345	...
1631	[name]	[address]	12345	...
1648	[name]	[address]	12345	...
1842	[name]	[address]	12345	...
2055	[name]	[address]	12345	...
2139	[name]	[address]	12345	...

This partitioning strategy can help reduce data access contention across different parts of a system.

Tool Data Partitioning

- Spark
- Hive
- Hadoop

Technique of hive Partition

- Managed Partitioned Table
- External Partitioned Tables
- Static Partitioning in Hive
- Dynamic Partitioning in Hive