Smarten
Augmented Analytics

Data Partitioning Guidelines

Document Version: 1.0
Product Version: 5.3

www.Smarten.com

Document Information	
Document ID	Smarten-Data-Partitioning-Guidelines
Document Version	1.0
Product Version	5.3
Date	10-February-2022
Recipient	NA
Author	EMTPL

Statement of Confidentiality, Disclaimer and Copyright

Disclaimer

This document suggests overall guidelines for configuring dataset and BI object partition columns. The absolute numbers given for such suggestions as partition size or the number of partitions should be treated as a general guideline, as performance will depend on the actual resources—CPU, memory available, and type of data storage on the server.

Contents

1 Introduction

Data Partitioning is the technique of distributing data across multiple files, tables, or disks to improve query processing performance or increase data manageability. Data partitioning enables parallel data processing. When data is partitioned across multiple disks or files, IO parallelism and in some cases query parallelism can be attained, as different partitions can be accessed and processed in parallel.

How multiple Partitions improve Performance

- Reduces the number of record reads based on given query criteria for key columns and increases query performance for large amounts of data
- Enables parallelism on reading and writing data
- Improves IO efficiency, as data is divided in multiple, relatively smaller files

How multiple Partitions deteriorate Performance

- It is important to create partitions aligned with resources available on the server and overall data size; otherwise, it will overload processing, and performance will diminish.
- For a smaller amount of data, it is not advisable to partition data because you are increasing processing overload that may not be required.
- If data is not partitioned based on search or filter columns in your data objects, such as reports, it may not produce the expected performance.

Factors to be considered when deciding partition columns for datasets and front-end objects data

- Number of partitions
- Total data size—number of records
- Ratio of data size and number of partitions
- Number of CPU cores available
- Memory available
- Number of partition columns
- Amount of data in each partition

How data is partitioned

This section explains how data is partitioned based on selected partition columns.

- If single column is selected as partition column, partitions will be created based on unique values of that particular column. For example, if “Year” column is selected as partition column and there are 5 years of data, it will create 5 partitions.
- If more than one column are selected as partition columns, number of partitions will be created based on distinct combinations of all selected columns data. For example, If “Year” and “Month” columns are selected and there are 5 years and each year has 12 months data, it will create 60 partitions.

2 Guidelines to decide Partition Columns

- **Define Partitions with Similar Data Size**
  If you have data for November 2020 through December 2021 and you are partitioning the data by the "year" column, it would not improve performance, as the partition 2020 has a low amount of data, and major data is under the 2021 partition, so it is advisable to partition on the column that can divide data equally, i.e., the "Month" column in this case.

- **Avoid Partitions with Minimal or Small Data Size**
  If you have data for November 2020 through December 2021 and you are partitioning on the "Transaction Date" column, it would not improve performance, as the number of records per partition will be quite low, and the total number of partitions will be high.
  It is recommended that the partitions' file size should be between 512 and 1024 MB.

- **Avoid creating too many Partitions**
  More partitions will result in more IO operations. As IO operations are heavy in performance, avoid a very large number of partitions, e.g., if you have 100 M records of data and you are partitioning on a unique key column, i.e., TransactionId, it will create 100 M partitions, so while reading the data, the system has to read 100 M files, which would take a great many IO operations and affect reading performance.

- **Keep a balance between the number of Partitions and the Average Data Size of Partitions**
  You should maintain a balance between the number of partitions and the data size for each partition, e.g., if your total data size is limited and you are creating too many partitions, it would not be beneficial. Also, if your total data size is very high and the number of partitions is too low, partitioning would not create an advantage. So, you should select partitions so that the partition file size can be between 512 MB and 1 GB.

- **Create the number of Partitions and the Data Size of Partitions based on available resources**
  You should create the number of partitions based on the number of Cores and Memory available on the server, e.g., if your server has 32 cores and 4 partitions, you would harness the available processing power of CPUs and not benefit from parallelism using all cores. Instead, data with 32 partitions would have more parallelism in processing data, as you would be utilizing the maximum number of cores available for processing.

- **More Partitions—more Files—more IO and more processing are required to Aggregate and Process Data**
  You should avoid creating too many partitions, which can make data reading less efficient with too many file reads, and the data consolidation process also requires more processing power, e.g., if you are creating a great many partitions when processing data, the system has to read a great many files, which includes a very long IO operation time compared with actual data processing time.

- **Fewer Partitions—fewer Files—not benefiting from processing in chunk through Partitions**
  You should avoid creating too few partitions. This will not give any advantage of parallelism and may not improve performance, e.g., if your server has 32 cores and you have created 2 partitions for your data, it will not utilize your server processing power, and you will not be using the processing resources available and may not be able to benefit from parallelism. You should create the number of partitions depending on your data size and cores available.

3 Data Partitioning decision and its effect on Performance

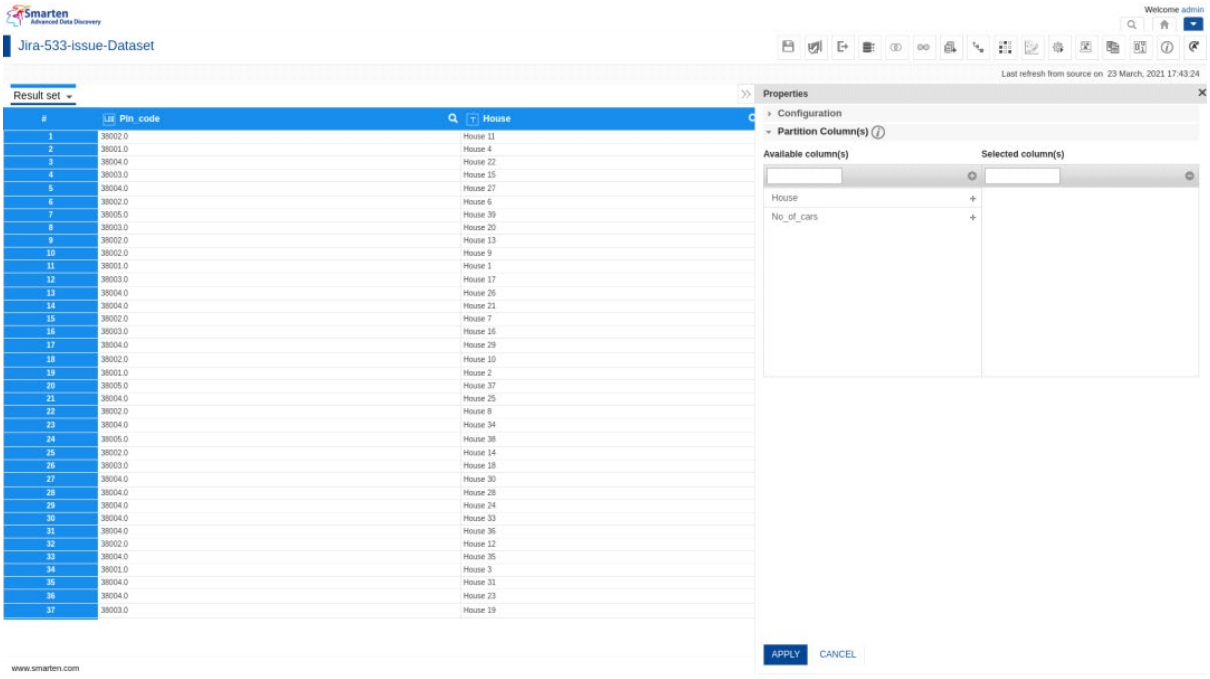
In this section, data partitioning scenarios and their effect on performance are explained.

Scenarios	Performance Impact
Selecting partition columns with low unique values (< 100)	Less unique values will create less number of partitions. It will improve IO operations and give better performance.
Selecting partition columns with high unique values (more than 1000 values)	More unique values will create more partitions. It will require more IO operations and CPU utilization to read and consolidate data. It will decrease the performance.
Apply partitioning with a low number of records per partition (less than 500K)	If there is less number of records per partition, read operation efficiency will be low and it will decrease performance.
Apply partitioning with a high number of records per partition	If number of records per partition is high and number partitions are same as CPU cores, it will give best IO and CPU performance and give good read performance.
Apply partitioning when fewer cores are available and the number of partitions is high	If available cores are less and number of partitions is too high, it will give more load to CPU and decrease read performance.
Apply partitioning when the number of cores available is high and the number of partitions is low	If number of cores available are high and number of partitions are less, it will not utilize full CPU power and it will not gain data partitioning benefits.
Apply partitioning when the number of partitions is high and the number of records per partition is low (less than 200K)	If there is less number of records per partition and number of partitions is high, IO operation and CPU operation efficiency will be low and it will decrease performance.
Apply partitioning when the number of partitions is low and the number of records per partition is very high	If number of partitions is low and number of records per partition is very high, it will do intense IO operations and under utilize CPU power. This will decrease the reading performance.

> Note:
> Partitions will be created based on distinct combinations of all selected partition columns data.

4 Partitioning on Datasets

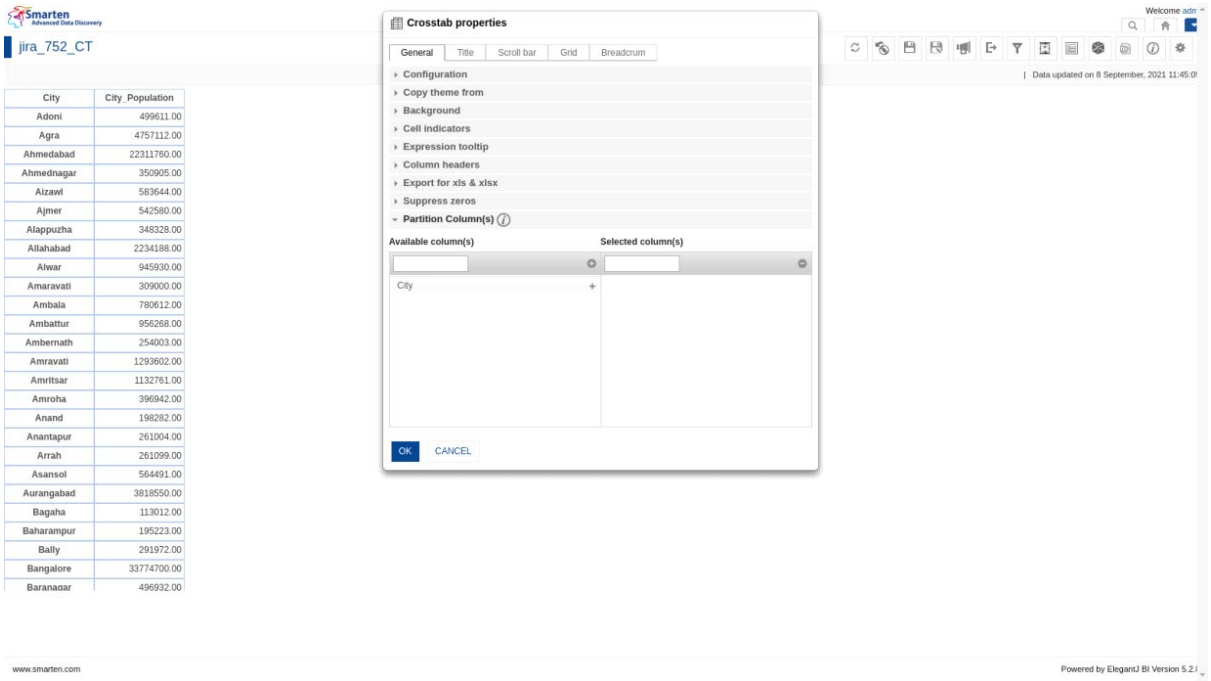
Smarten supports data partitioning for cache datasets. Users can select partition columns for datasets from dataset properties. Dataset published data is partitioned as per selected partition columns. This configuration improves data reading performance when reading from dataset data is required in such actions as creating a new BI object, changing a BI object outliner, using datasets in blend operations, or drilling through from BI objects. When a dataset is enabled for managed memory, the benefit of data partitioning is applicable only when dataset data is being loaded into memory from a disk. Data partitioning is not applicable for real-time datasets.



DATA PARTITIONING—DATASET

5 Partitioning on Front-end BI Objects Data

Smarten also supports data partitioning for front-end BI objects. Users can partition BI object data by selecting partition columns from BI object properties. BI object data is partitioned as per selected partition columns. This feature improves data reading performance when processing data for the BI objects, so it helps improve performance in such actions as opening and processing BI objects, applying filters on BI objects, or drill down actions. When a BI object is enabled for managed memory, the benefit of data partitioning is only applicable when data is being loaded into memory from a disk. The data partitioning option is not applicable for BI objects that are created from real-time datasets.



DATA PARTITIONING—BI OBJECTS

6 Product and Support Information

Find more information about Smarten and its features at www.smarten.com
Support: support@smarten.com
Sales: sales@smarten.com
Feedback & Suggestions: support@smarten.com
Support & Knowledgebase Portal: support.smarten.com