Why Partition a Table

ClustrixDB's distributed database architecture automatically handles many use cases for which legacy RDBMS applications required table partitions.

Here are some typical scenarios for partitioned tables and see if and how they are handled by ClustrixDB.

When to use Partitioned tables with ClustrixDB

Purging old data

Partitioned tables are often used to expedite loading, purging, or archiving of dated information. For example, one might partition a table by month and drop the oldest month's data at the beginning of each new month.

With ClustrixDB, dropping partitions is more efficient than deleting data. EXCHANGE PARTITION can be used to roll archive data to a historical table.

When Partitioned tables are no longer needed

Since ClustrixDB automatically slices and distributes data across the cluster and automatically applies parallelism, many use cases that require partitions for on legacy systems are not required or recommended with ClustrixDB.

Legacy usage of partitioned tables (other RDBMS)	Why this is unnecessary in ClustrixDB	
For better query performance from partition elimination or partition pruning.	Only partitions that contain potential matching values for a query are accessed. ClustrixDB distributes data and indexes into slices and uses slices to ensure that only relevant data is accessed. Table partitioning provides no additional query performance benefit for ClustrixDB.	
Table partitioning to utilize multiple disks.	Large tables can be spread across multiple partitions over multiple disks to reduce disk contention. ClustrixDB automatically distributes data slices across multiple nodes of the cluster using its Rebalancer. The Rebalancer automatically arranges data throughout the cluster to lessen disk contention. Explicit table partitioning is not required to utilize multiple disks and does not provide any improvement to disk utilization.	
Partitioning tables to provide parallel execution.	ClustrixDB leverages distributed data slices and distributed query fragments for parallel execution on multiple nodes. Partitioning tables within ClustrixDB will provide no additional parallelism.	
Table partitioning to expedite data management.	Processes such as table indexing, backups, restores, etc. are streamlined when performed on subsets of partitioned tables.	