

Partitioned Tables Limitations

Limitations, caveats, and unsupported features for using Partitioned Tables in ClustrixDB.

Unsupported in Partitioned Tables

This is the list of specific MySQL Partitioned Table features that are unsupported in ClustrixDB.

- LIST, HASH, KEY, LINEAR KEY partitions
- SUBPARTITIONS
- Partitioning across functions such as: ABS(), DAYOFMONTH(), DAYOFWEEK(), DAYOFYEAR(), DATEDIFF(), EXTRACT(), MICROSECOND(), MINUTE(), MOD(), MONTH(), HOUR(), SECOND(), QUARTER(), TIME_TO_SEC(), WEEKDAY()
- Temporary tables may not be partitioned
- COALESCE PARTITION number
- ANALYZE PARTITION {partition_names | ALL }
- CHECK PARTITION {partition_names | ALL }
- OPTIMIZE PARTITION {partition_names | ALL }
- REBUILD PARTITION {partition_names | ALL }
- REPAIR PARTITION {partition_names | ALL }
- EXPLAIN PARTITION
- SELECT FROM table_name PARTITION {partition_name | ALL }

Caveats of Using Partitioned Tables

- Clustrix recommends no more than 64 partitions per table. The maximum number permitted is configurable via the global variable max_partitions.
- To determine the number of slices utilized by a partition, SHOW FULL CREATE TABLE and count the number of HASHES.
- A partition's ordinal position (the consecutive sequence in which a partition was added to the database) is unavailable in ClustrixDB.
- Adding data to a table that does not fit within any partition will cause this error

```
ERROR 1526 (HY000): [6147] Table has no partition for value: nnn
```

- Altering a partitioned table and modifying characteristics of a column used for partitioning requires that you respecify partitioning ranges for the table. For example:

```
sql> /* Given this table definition: */
sql> CREATE TABLE test (id INT KEY)
      PARTITION BY RANGE (id)
      (PARTITION p0 VALUES LESS THAN (1000),
       PARTITION p1 VALUES LESS THAN (2000));
Query OK, 0 rows affected (0.19 sec)

sql> /* Changing characteristics of the partitioned column causes this ALTER to fail: */
sql> ALTER TABLE test
      MODIFY COLUMN id INT AUTO_INCREMENT;
ERROR 1 (HY000): [12298] Invalid constraint specification: Index
test.__idx_test__PRIMARY on a partitioned table must contain the partition column

sql> /* But respecifying the partitioning details allows this ALTER to succeed: */
sql> ALTER TABLE test
      MODIFY COLUMN id INT AUTO_INCREMENT
      PARTITION BY RANGE (id)
      (PARTITION p0 VALUES LESS THAN (1000),
       PARTITION p1 VALUES LESS THAN (2000));
Query OK, 0 rows affected (3.98 sec)
```

- If partitioning using a partition key that includes the YEAR datatype, you must also use the YEAR function as shown in this example:

```
sql> CREATE TABLE sample_year_partitioning (year_col YEAR PRIMARY KEY)
      PARTITION BY RANGE (YEAR(year_col))
      (PARTITION p0 VALUES LESS THAN (5),
       PARTITION p1 VALUES LESS THAN (10),
       PARTITION p2 VALUES LESS THAN MAXVALUE);
```