REPLICAS

ClustrixDB maintains multiple copies (replicas) of each slice of data to provide fault tolerance, high availability, and ensure reads are balanced. There are at least two physical replicas of each logical slice, stored on separate nodes (or zones). Writes are applied to all replicas simultaneously, but a single replica is designated as the "ranked replica" and is used for reads.This helps to keep the distribution of reads even across the cluster. The rebalancer rerank process continuously assesses cluster load balance for reads, and as necessary, may re-designate the ranked replica of a given slice to keep load even.

The following rules determine replica placement within the cluster:

- By default, there are two copies of each slice (replicas = 2)
- Replicas are distributed across the cluster for redundancy and to balance reads, writes, and disk usage.
- No two replicas for the same slice will exist on the same node or in the same zone.
- ClustrixDB will make new replicas while the database remains online, without suspending or blocking writes to the slice.

Specifying the Number of Replicas

When creating a table, by default the number of replicas created is based on the global variable MAX_FAILURES. The default MAX_FAILURES = 1 results in REPLICAS = 2.

Once a table has been created, you can use an ALTER statement to modify the number of replicas:

```
ALTER TABLE
tbl_name [RE
PLICAS = n]
```

The default number of replicas as specified by MAX_FAILURES is appropriate for the majority of use cases. The number of replicas is also configurable by representation. For example, a user may require three replicas for the base representation of a table and only two replicas for the other representations of that table.

For most workloads, the default of 2 replicas is optimal for balancing fault tolerance with performance. Setting REPLICAS = 1 will remove fault tolerance and is not recommended.

To configure the number of replicas for a specific representation at table creation:

```
CREATE TABLE tbl_n
ame col_names
[REPLICAS = n]
```

This will automatically copy your sliced data to multiple nodes until the desired number of replicas are created.

Configuring additional replicas is not sufficient to ensure fault tolerance in the face of multiple failures. For more information on configuring your cluster for multi-node failures, see MAX_FAILURES.

ALLNODES

REPLICAS = ALLNODES specifies that a complete copy of the table is maintained on every node. When ALLNODES is used, ClustrixDB is able to utilize local copies of the table to more efficiently execute queries. This is especially applicable when small tables (ALLNODES) are joined with larger tables. However, maintaining these copies has a significant overhead to write performance.

ALLNODES is best used for tables that meet the following criteria:

- Relatively small (Table Size < 10MiB)
- Written to infrequently (Write Frequency < 1K)
- Read from frequently (Read Frequency > 1M)
- Used frequently in joins to other, larger tables (e.g. metadata, lookup tables)

ALLNODES should not be used for:

- Tables that take frequent writes
- Partitioned tables

REPLICAS = ALLNODES Syntax

```
CREATE TABLE tbl_name
(col_names) [REPLICAS = A
LLNODES]
ALTER  TABLE tbl_name
[REPLICAS = ALLNODES]
```