ClustrixDB System Table Reference

This topic describes the tables that comprise the ClustrixDB system database.

System tables cannot be altered by users. The following is informational only.

ALTER_PROGRESS

The alter_progress table provides a rough estimate of the progress for an ALTER.

Name	Description
db	The ID for the session.
src_relation_name	The name of the source relation.
start	Timestamp for when the alter started.
rows_read	The number of rows read by the ALTER.
src_relation_est_rows	The estimated number of rows in the source relation
completion_est	A rough estimate of when the ALTER might finish.

DATABASES

The **databases** table contains one row for each database in the system. Its columns are as follows:

Name	Description
db	Contains an ID for each database. It corresponds to the **db** column in the **relations** table.
user	Contains the ID of the user who created the database. It corresponds to the **user** column of the **users** table.
name	Name of the database.
hidden	Indicates whether the table is hidden from users when performing queries. 1 = yes, 0 = no.

cscl	An internal value that encodes the character set and collation sequence utilized by a given database.
	To translate this encoded value to English, a special function is available as follows:
	select OIDTYPE_CSCL_PROP(cscl, 'charset') from system.databases where name = 'sample';
	select OIDTYPE_CSCL_PROP(cscl, 'collation') from system.databases where name = 'sample';

CLUSTER_SESSION_STATS

The **cluster_session_stats** table contains per-session statistics about transactions and statement types.

GTM_COORD

This table contains a row for each open transaction.

MYSQL_MASTER_STATUS

The **mysql_master_status** table stores information about each replication slave that is currently connected to the cluster.

REBALANCER_SUMMARY

The cluster performs ongoing optimization tasks in background mode. In rare cases, this background activity might affect query performance. In such cases, the Rebalancer, which moves data between disks and nodes to ensure even distribution, is a likely source. Rebalancing typically runs after an import, and occasionally runs as disk utilization increases. Because Rebalancer sessions can be brief, the **rebalancer_activity_log** table is a better source for detailed information.

Values returned in the **action** column are as follows:

Action	Description
split	A slice has grown too large and is being split into two smaller slices.
move	A slice is being moved to another disk or node to optimize the load.
copy	A slice is being duplicated, usually during the reprotection stage after a hardware failure.

REBALANCER_ACTIVITY_LOG

This table contains a detailed log of Rebalancer actions and the reasons for those actions. See Monitoring Data Rebalancing Activity.

RELATIONS

The **relations** table contains one row for each table in the system.

REPRESENTATIONS

The **representations** table contains one row per representation in the system. Each relation and each index on that relation constitutes a representation. The **base** column indicates when the representation is the Base Representation.

SESSIONS

The **sessions** table contains information about all currently connected and recently disconnected client sessions.

STATS

The **stats** table contains internally-collected statistics about database and cluster performance. Most of these statistics are for internal use, but some are of general interest.

TABLE_REPLICAS

The **table_replicas** table contains one row for each replica in the system and records each replica's size and node. This table can be used to determine the total calculated size of an index and to find poorly-distributed indexes. Each slice is replicated two or more times and each replica resides on a different node. Each replica of a slice contains the same data.

TABLE_SIZES

The **table_sizes** table records the total size of each table in bytes, including its base representation and all of its indexes.

TABLE_SLICES

Representations are split into slices and distributed throughout your cluster. The **table_slices** table enumerates those slices and indexes and contains one row for each slice in each table.

Each slice contains a unique set of rows from the table. The complete table is represented by assembling all the rows from the table slices. The rows are allocated to slices using a hash that is based on the row's index or indices. The number of slices per table depends on many factors including the size of the cluster. To configure this setting, issue the ALTER TABLE SET SLICES = <N> command. Changing the slice count can improve performance. For details, contact Clustrix Support.

To display the names of the database, table, and indexes for each slice, join on the **slice** column.

VERSION_HISTORY

The version_history table provides a history of installation and upgrade versions.

USERS

Database user, host, and password combinations are stored in the **users** table. Join to this table to display user names instead of user ids. See also Managing Users and Privileges and Migrating Users and Permissions.