

Creating table

```
CREATE TABLE Test (
  Id INTEGER AUTO_INCREMENT,
  IntField INTEGER NOT NULL DEFAULT 0,
  PRIMARY KEY(Id))
WITH SYSTEM VERSIONING;
```

Creating table - partitioned

```
CREATE TABLE Test (
  Id INTEGER AUTO_INCREMENT,
  IntField INTEGER NOT NULL DEFAULT 0,
  PRIMARY KEY(Id))
WITH SYSTEM VERSIONING
PARTITION BY SYSTEM_TIME (
  PARTITION PART_HISTORY HISTORY,
  PARTITION PART_CURRENT CURRENT
);
```

This partitions the table based on the current and historical data. This can boost performance a lot.

Creating table - Skip field from versioning

```
CREATE TABLE Test (
  Id INTEGER AUTO_INCREMENT,
  IntField INTEGER NOT NULL DEFAULT 0,
  CharField VARCHAR(255) NOT NULL DEFAULT "",
  PRIMARY KEY(Id))
WITH SYSTEM VERSIONING
PARTITION BY SYSTEM_TIME (
  PARTITION PART_HISTORY HISTORY,
  PARTITION PART_CURRENT CURRENT
);
```

When you update **only** the CharField field the server won't save a new version of this row.

Select data

```
SELECT * FROM Test
FOR SYSTEM_TIME ALL;
```

This will select all rows with all saved versions.

Select data with rows lifespan

```
SELECT *, ROW_START, ROW_END FROM Test
FOR SYSTEM_TIME ALL;
```

The **ROW_START** and **ROW_END** fields are invisible columns. These are representing the time when the row has that data.

Select data at a time

```
SELECT * FROM Test
FOR SYSTEM_TIME AS OF TIMESTAMP '2018-04-13 07:13:00';
```

```
SELECT * FROM Test
FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 YEAR)
AND NOW();
```

```
SELECT * FROM Test
FOR SYSTEM_TIME
FROM '2018-04-13 00:00:00'
TO '2018-07-13 00:00:00';
```

Adding or removing system versioning

To add system versioning to a table simply do:

```
ALTER TABLE Test ADD SYSTEM VERSIONING;
```

To remove the system versioning do this:

If you used partitioning, first you need to remove these:

```
ALTER TABLE Test REMOVE PARTITIONING;
```

Then drop the versioning info:

```
ALTER TABLE Test DROP SYSTEM VERSIONING;
```

Warning: this will remove all historical data from this table!

Warnings

The system versioned tables cannot be truncated. Simply drop and recreate them if needed.

To alter a system versioned table you must set **system_versioning_alter_history** to **KEEP** in your MariaDB config file (my.cnf or my.ini) in the [mysqld] section.

mysqldump cannot read versioned data.

You must use MariaDB backup to make a backup of this kind of data.

