

# Rebalance compaction

In order to improve performance, Hive under the hood creates bucket files even for non-explicitly bucketed tables. Depending on the usage, the data loaded into these non-explicitly bucketed full-acid ORC tables may lead to unbalanced distribution, where some of the buckets are much larger (> 100 times) than the others. Unbalanced tables has performance penalty, as larger buckets takes more time to read. Rebalance compaction addresses this issue by equally redistributing the data among the implicit bucket files.



- Rebalance compaction is never initiated automatically.
- Unlike other compaction types, rebalance compaction puts an *exclusive write lock* on the table.

SQL example:

```
ALTER TABLE table_name COMPACT 'REBALANCE';
```

## Number of implicit buckets

It is possible to set the desired number of implicit buckets during a rebalance compaction.

SQL example:

```
ALTER TABLE table_name COMPACT 'REBALANCE' CLUSTERED INTO n BUCKETS;
```

If the bucket number is not supplied, the number of buckets will remain the same, only the data will be redistributed among them.

## Ordering

Optionally, an order by expression can be supplied, to be able to re-order the data during the rebalance.

SQL example:

```
ALTER TABLE table_name COMPACT 'REBALANCE' ORDER BY column_name DESC;
```

If the *order by* expression is not set, the ordering of the data remains the same after the compaction.

## Limitations



- Rebalance compaction can be done only *within partitions*.
- Rebalance compaction is supported only on *implicitly bucketed* tables, clustered tables are not supported.
- Rebalance compaction is supported only on *full-acid* tables, insert-only tables are not supported.
- Rebalance compaction is possible only via *query-based compaction*, MR based compaction is not supported.