Bigtable is a distributed storage system developed internally by Google to handle petabyte-scale applications - like Google Maps and Google Analytics. In 2015 Bigtable was launched as a public service on Google Cloud.

Similar to many other datastores, Bigtable uses a structure of rows and columns. Where it differs is in that Bigtable only allows for a single index - the row-key. Three primary operations exist:

* **Mutate** - used to change the value connected to a row key and also for inserts.
* **Read** - read the value of a single row key.
* **Range** - read a range of row keys. Both filtering and aggregations are supported.

To achieve scaling, Bigtable splits the data into *nodes*. Each *node* is a commodity server and stores a range of row keys. As a result, choosing the right key for your workload is crucial.

For example, consider the example of storing a marketing event in Bigtable. The event looks like this:

<https://gist.githubusercontent.com/halvorboe/4dd8248bc1b8544266d02560075e296e/raw/19bdc024ea396fb8fadc5800cd888afea37e3d0c/event.json>

There are three types of data here:

## Random

The row key is random. Random data should not be the beginning if a low key in Bigquery because it makes range queries useless.

## Timestamp

The timestamp for the example is a long time ago, but when dealing with timestamps, it is essential to get them in the right format.

2019#12#03#14:05:12

## URL

We are reversing the most common.

In summary, Bigtable is a great datastore - if you are storing massive amounts of data. Otherwise, it's terrible.