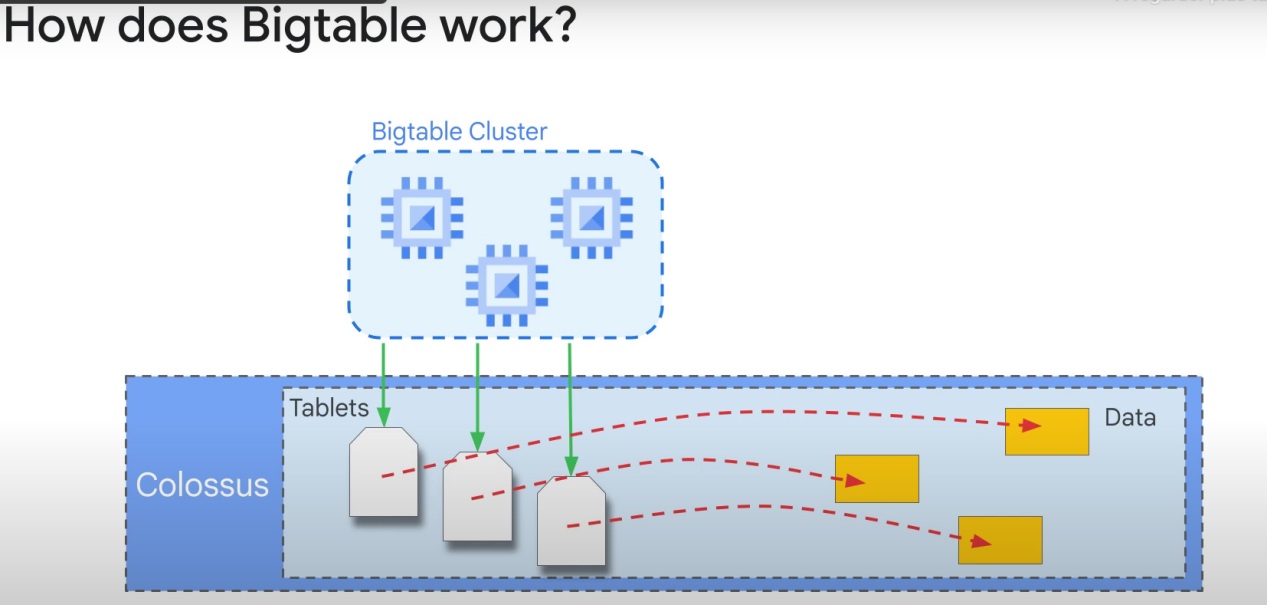


BigTable is not good with joins and data volume of less than 1TB

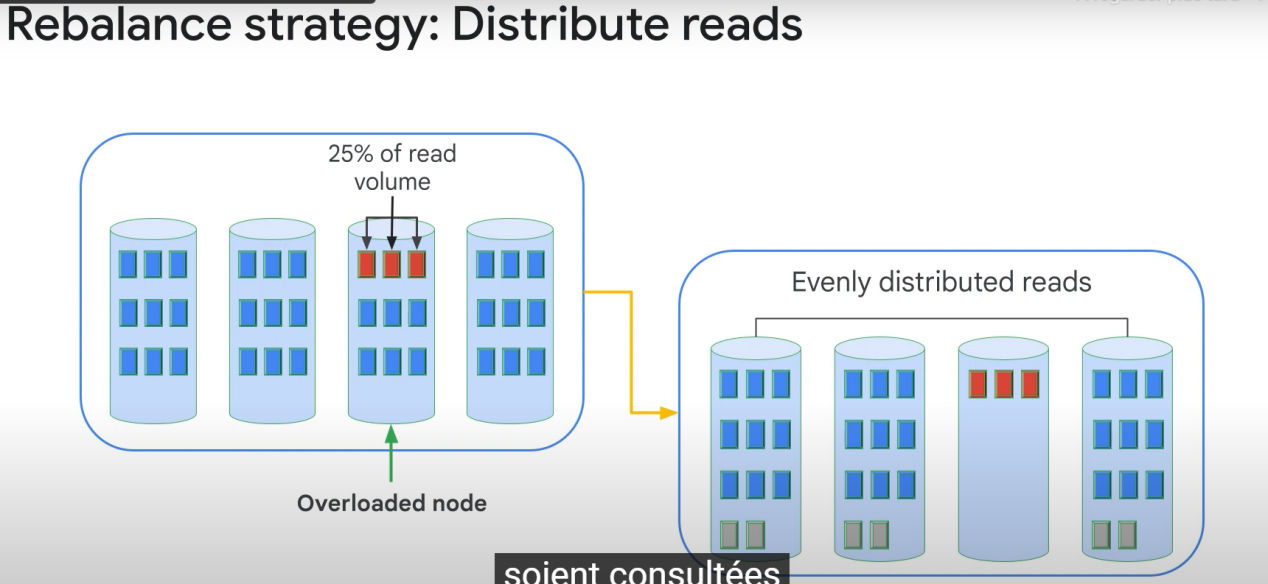


Data is stored in a file system called Colossus.

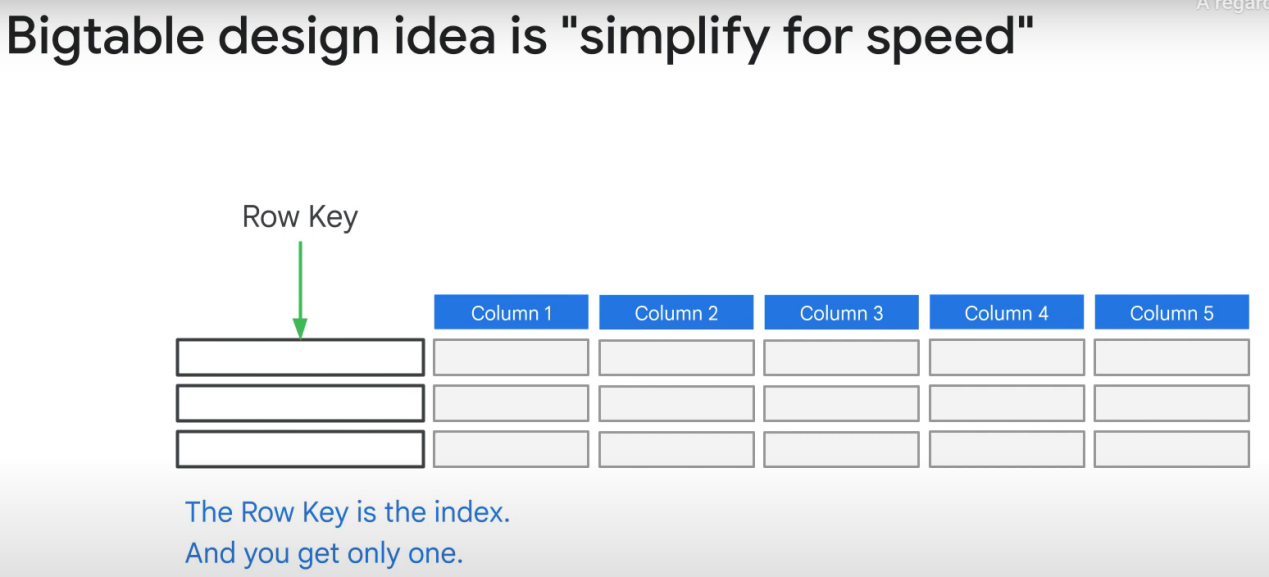
Each tablet is used to identify and manage the data. The metadata that points to the tablets is what is stored in the clusters.

The strength of BigTable is that it can rebalance the processes by moving the pointer to a tablet to a different VM in a cluster. When we have constant use of data over a long period of time, this gives time to bigTable to figure out the best possible repartition.

Read and write operations are distributed evenly across an entire table and cluster

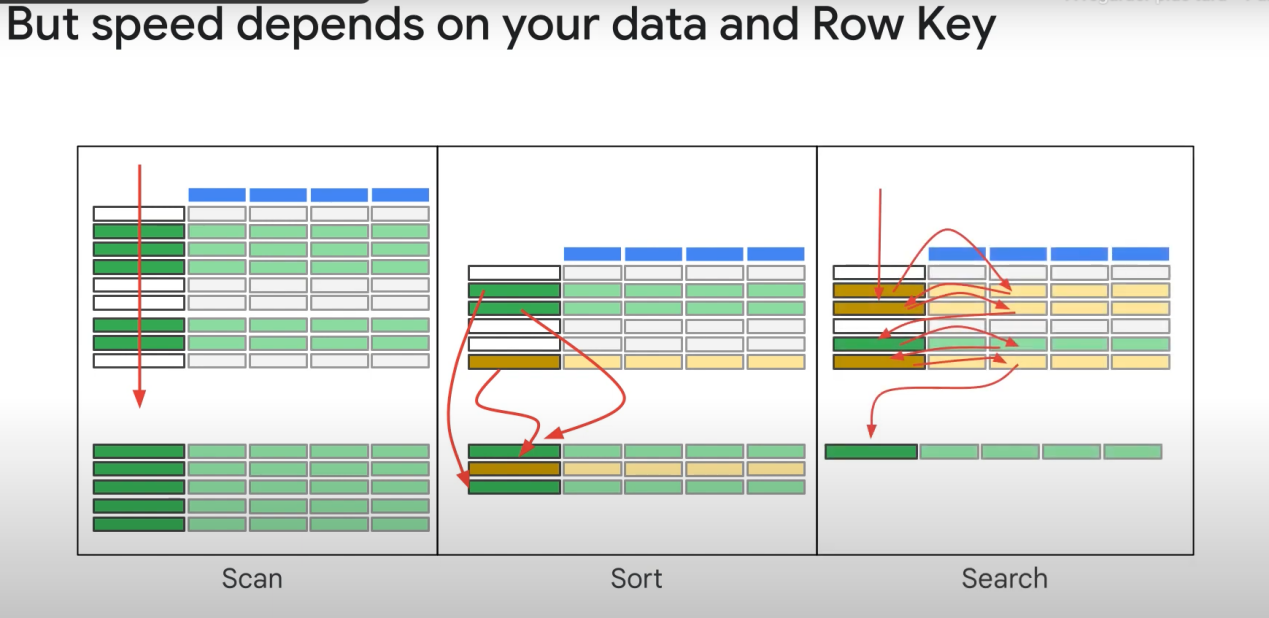


Equally distributing readsoperations has taken priority to distributing storage across the cluster.



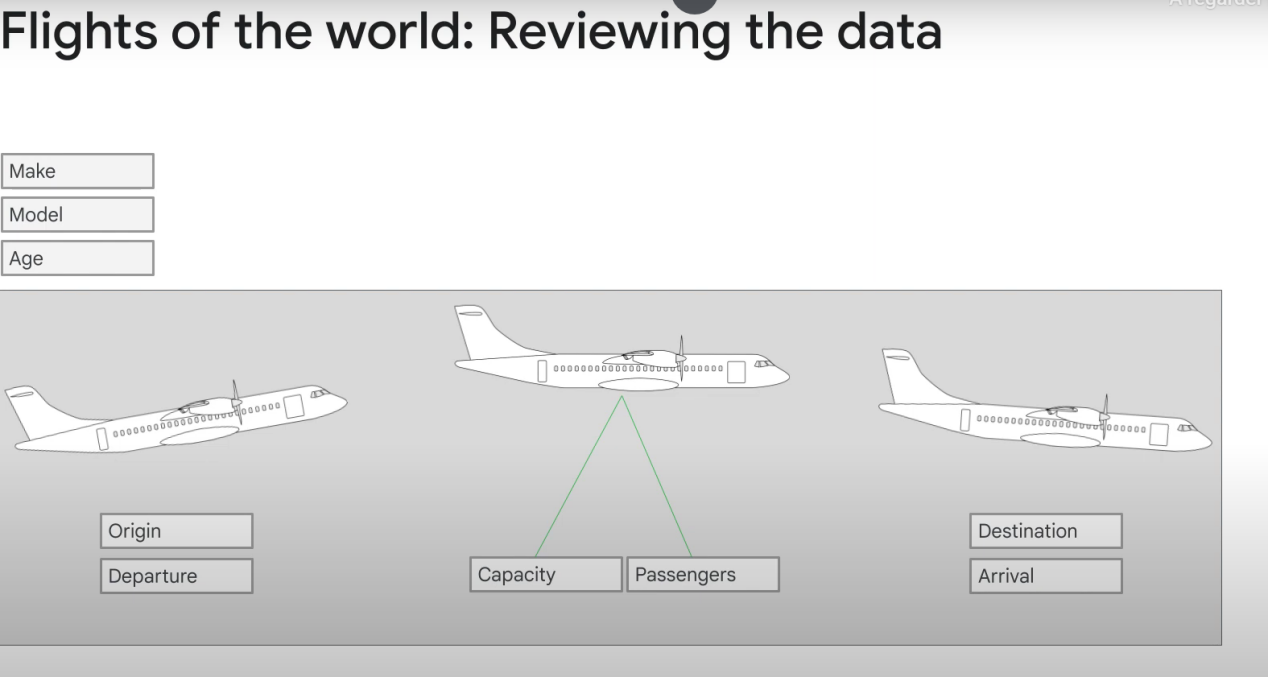
Data is organized by the row key. No other secondary index.

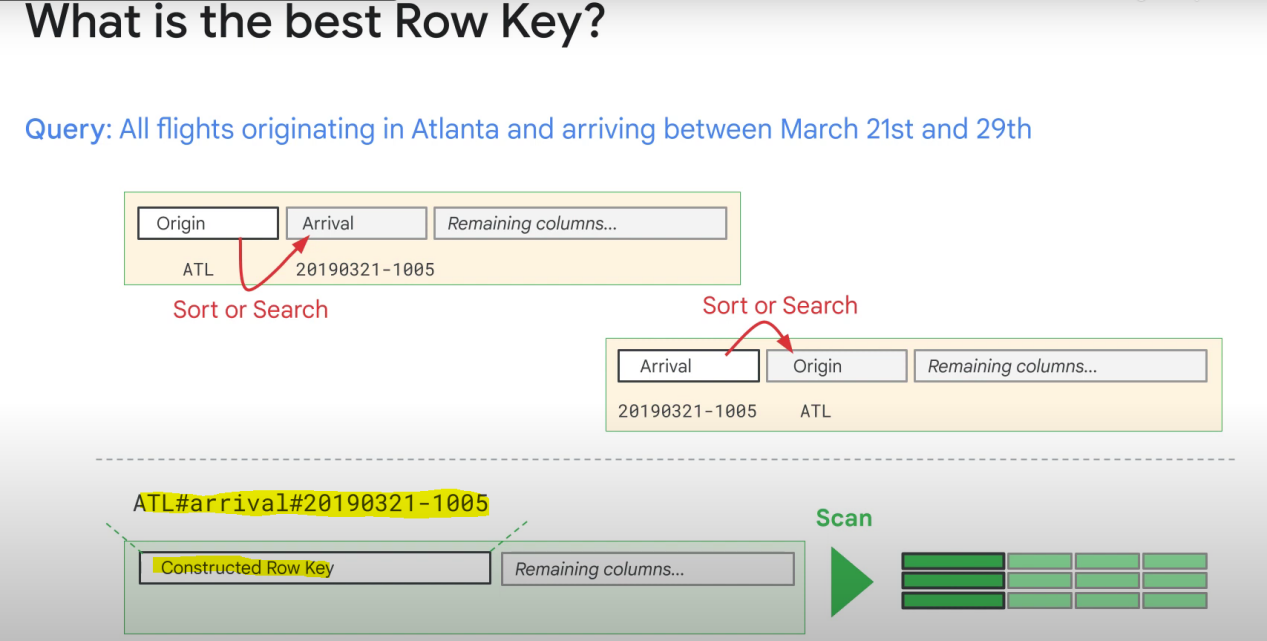
Big table is all about simplicity. It doesn’t use SQL (too many operators/commands). It’s considered noSql.



For sort, we do 1 scan and then sort the intermediary result. The sorting operation will take less time if most of the rows are already kind of sorted.

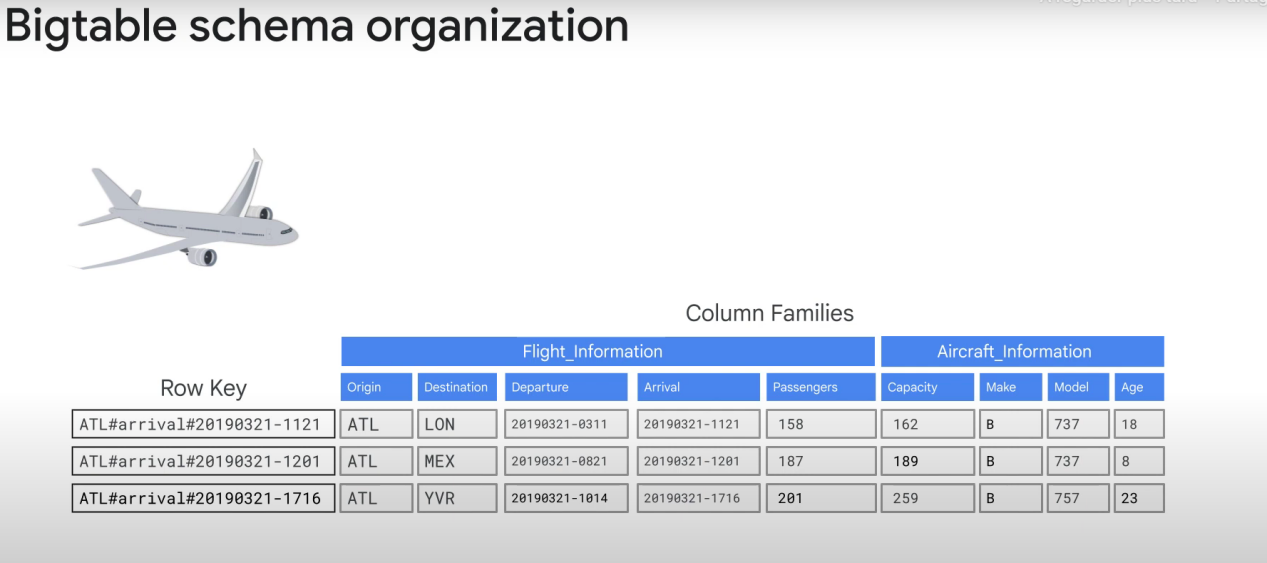
Choosing the best row key (index)



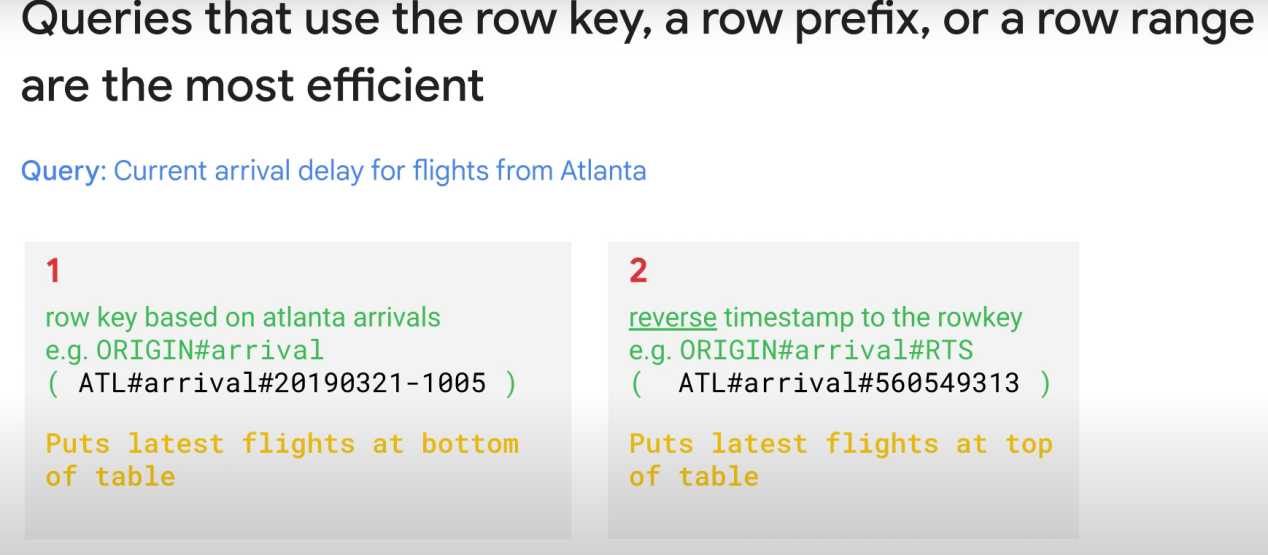


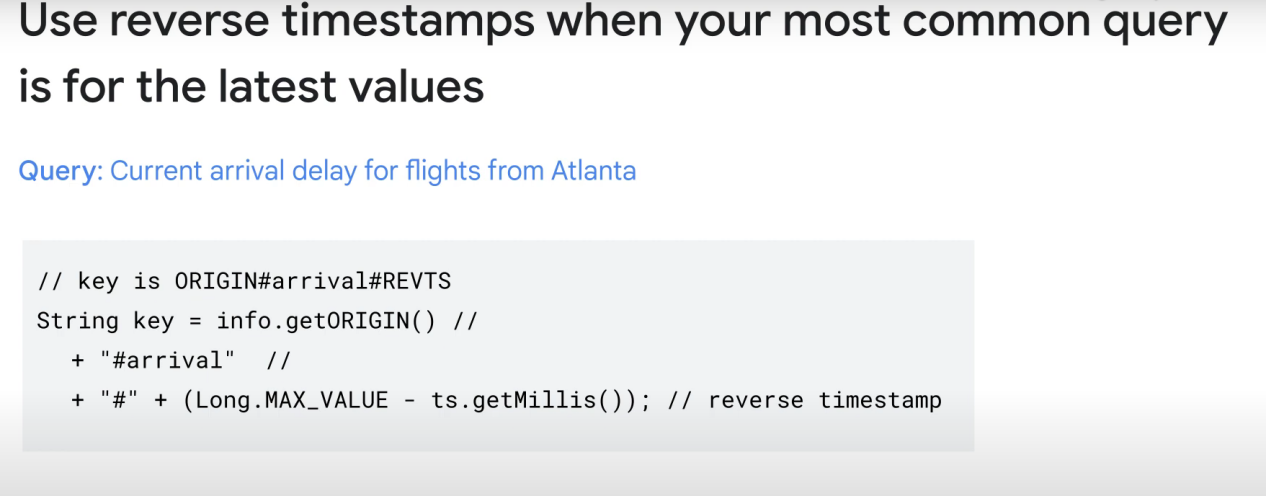
We can create our own custom key! And because the rows are stored lexicographically by the row key, we can easily generate the solution set with 1 scan.

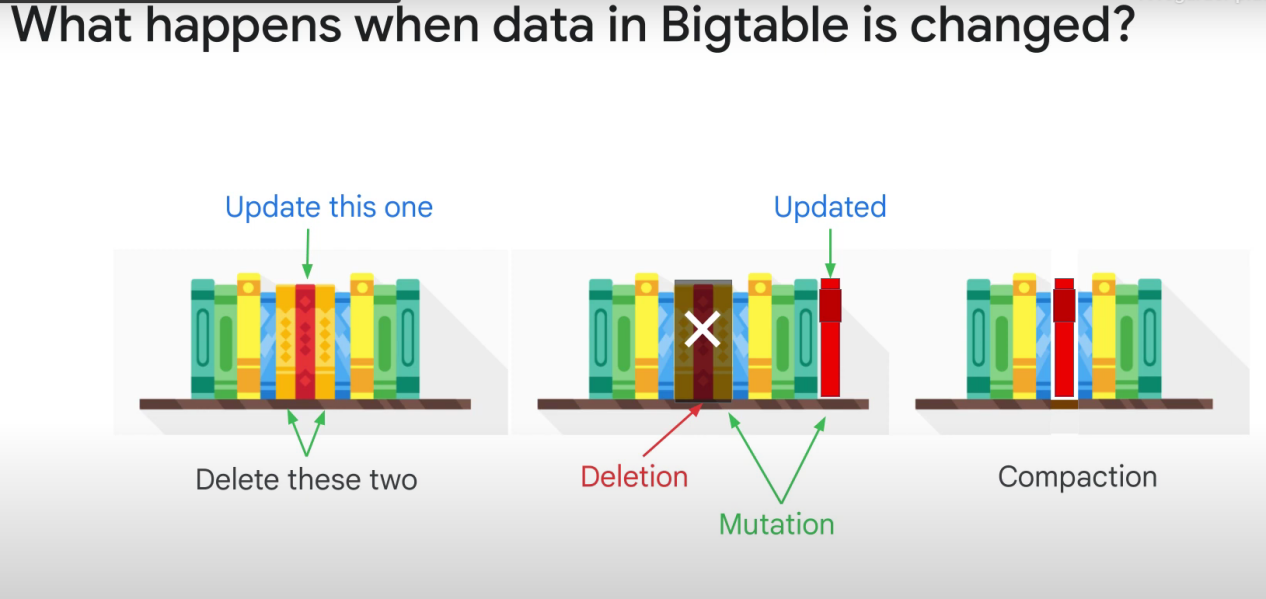
We can create column families, so that we only look at those instead of having to load the entire row.



BigQuery can handle up to 100 **column families** without having performance problems.







When you delete a row, it is marked for deletion and not taken into account for any incoming query, but it is not immediately removed.

If you make a change in a row, the «new row» is appended to the end of the table and the old one is marked for deletion, but not instantly deleted. Both rows exist for a period of time.

Periodically, bigquery reorganizes the data and deletes the rows marked for deletion.