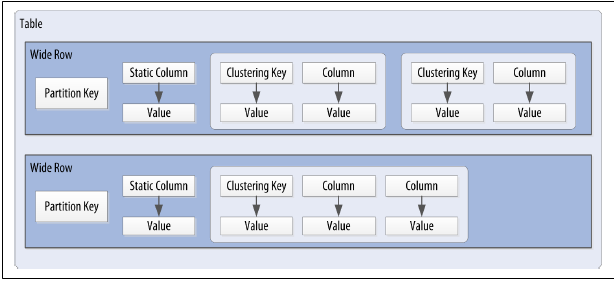
primary key:

* **partition key** (The partition key is used to determine the nodes on which rows are stored)
* **clustering key** (used to control how data is sorted for storage within a partition)



Cassandra data structures:

* The **column**, which is a name/value pair
* The **row**, which is a container for columns referenced by a primary key
* The **table**, which is a container for rows
* The **keyspace**, which is a container for tables
* The **cluster** or **ring**, which is a container for keyspaces that spans one or more nodes

Timestamps:

Each time you write data into Cassandra, a timestamp is generated for each column value that is updated.

* (Why) Cassandra uses these timestamps for resolving any conflicting changes that are made to the same value. Generally, the last timestamp wins.

**Points to remember:**

- writetime() function is used to display the timestamps value.

- USING TIMESTAMP option to manually set a timestamps

- It is not allowed to set the timestamps value of a primary key column.

**Use case of manually set timestamps in Cassandra:**

Setting the time-stamp is not required for writes. However, this functionality is typically used for writes in which there is a concern that some of the writes may cause fresh data to be overwritten with stale data (e.g. from a test data load process triggered automatically). This is advanced behavior and should be used with caution.

Time to live (TTL):

One very powerful feature that Cassandra provides is the ability to expire data that is no longer needed. The time to live (or TTL) is a value that Cassandra stores for each column value to indicate how long to keep the value.

**Points to remember:**

- TTL is stored on a per-column level.

- There is currently no mechanism for setting TTL at a row level directly

- The TTL value defaults to null

- TTL() function is used to display the value of TTL.

- USING TTL option is used to set the TTL value on a column.

- It is not allowed to set the TTL value of a primary key column.

**Use Case:**

TTL can be used to delete column(s) automatically from Cassandra database after a certain time instead of using delete command.

**Question?**

**Performance - TTL vs Deleting a row in Cassandra?**

* Deleting rows from Cassandra table need to run Job which may cause heavy load on the existing system. Whereas using TTL, we can automatically delete column values those are not required anymore. Therefore, performance wise, TTL is better approach.