

## Eventual Consistency:

Over time (if no new changes are made) each copy of the data will be the same, but if there are new changes, the data may be different in different locations. The data may be inconsistent for only milliseconds. There are workarounds in place to prevent getting stale data.

## Commonly Asked Questions:

### What does the network look like? Can you share any examples?

In Apache Cassandra every node is connected to every node -- it's peer to peer database architecture.

### Is data deployment strategy an important element of data modeling in Apache Cassandra?

Deployment strategies are a great topic, but have very little to do with data modeling. Developing deployment strategies focuses on determining how many clusters to create or determining how many nodes are needed. These are topics generally covered under database architecture, database deployment and operations, which we will not cover in this lesson. Here is a useful link to learn more about it for [Apache Cassandra](#).

In general, the size of your data and your data model can affect your deployment strategies. You need to think about how to create a cluster, how many nodes should be in that cluster, how to do the actual installation. More information about deployment strategies can be found on this [DataStax documentation page](#)

## Citation for above slides:

Here is the [Wikipedia page](#) cited in the slides.

## Cassandra Architecture

We are not going into a lot of details about the Apache Cassandra Architecture. However, if you would like to learn more about it for your job, here are some links that you may find useful.

### Apache Cassandra Data Architecture:

- [Understanding the architecture](#)
- [Cassandra Architecture](#)

The following link will go more in-depth about the Apache Cassandra Data Model, how Cassandra reads, writes, updates, and deletes data.

- [Cassandra Documentation](#)