# Understand this:

Eventual consistency:

1. I tell you that it's going to rain tomorrow.
2. Your neighbor tells his wife that it's going to be sunny tomorrow.
3. You tell your neighbor that it is going to rain tomorrow.

Eventually, all of the servers (you, me, your neighbor) know the truth (that it's going to rain tomorrow), but in the meantime the client (his wife) came away thinking it is going to be sunny, even though she asked after one of the servers knew the truth.

As opposed to Strict Consistency / ACID compliance:

1. Your bank balance is $50.
2. You deposit $100.
3. Your bank balance, queried from any ATM anywhere, is $150.
4. Your daughter withdraws $40 with your ATM card.
5. Your bank balance, queried from any ATM anywhere, is $110.

At no time can your balance reflect anything other than the actual sum of all of the transactions made on your account to that exact moment.

The ***reason*** why so many NoSQL systems have eventual consistency is that virtually all of them are designed to be distributed, and with fully distributed systems there is super-linear overhead to maintaining strict consistency (meaning you can only scale so far before things start to slow down, and when they do you need to throw exponentially more hardware at the problem to keep scaling).

[Source: StackOverflow](http://stackoverflow.com/questions/10078540/eventual-consistency-in-plain-english)

# Read this article:

[Next gen NoSQL: The demise of eventual consistency?](https://gigaom.com/2013/11/02/next-gen-nosql-the-demise-of-eventual-consistency/)

Read Wikipedia NoSql:

<https://en.wikipedia.org/wiki/NoSQL>

# Read this article:

[Balancing Strong and Eventual Consistency with Google Cloud Datastore](https://cloud.google.com/datastore/docs/articles/balancing-strong-and-eventual-consistency-with-google-cloud-datastore/)

Create a google drive document.

In the google drive document, provide explanations to the following:

* Describe the three ways of handling relational data in a NoSql database discussed in the wikipedia article.
* What is strong consistency?
* What is eventual consistency?
* According to the “Balancing strong and eventual consistency” article, how can memcache be used in conjunction with datastore to minify eventual consistency?