Homework - 1

Pick any 5 and answer briefly.  
We will pick up the best 5 if you answer all 6. Maximum points = 20

1. **Does C in CAP and ACID mean the same. Explain.**

Answer:

No. Even though both C in CAP and C in ACID are the abbreviation of “Consistency”, two consistencies have totally different meaning:

1. C in CAP means that as long as a transaction (like data is written) is executed, the dataset must be updated in all replications too. In other words, if a transaction is considered as “completed” or “successful”, all clients must be able to access the same data at the same time. In short, different nodes respond with the same data to the same request.
2. C in ACID describes a consistent state in the whole system or say, “internal consistency”. In order words, the data is in a consistent state from the transaction starts to the transaction commits. In short, it specifies the relationship and rules between different fields to keep the stable state for the whole dataset.
3. **NoSQL databases are easy because you need not worry about the schema. What do you think?**

Answer:

I think this statement is incorrect. Even though NoSQL database does not have a pre-defined and well-designed schema like SQL database, it has dynamic schema for unstructured data. The schema for NoSQL does not have predefined structure and it can be changed based on the underlying circumstances, so we need to worry about or say, “focus on”, the underlying schema.

1. **A closely held secret which people often overlook. Relational databases get into a bad state if 2 transactions are running simultaneously against the same object: for example if 2 different users are updating the same account (and data/records is all stored in a RDBMS), it can leave the data in a bad state. Is this true or False - please explain.**

Answer:

I think this statement is false. Data is stored in RDBMS so it will comply with ACID forever. I in ACID stands for isolation, meaning that the transactions are isolated when they execute concurrently, and they do not affect each other. Therefore, even though 2 transactions are running simultaneously against the same object, each transaction is unaware of the other transaction concurrently occurring in the system and the data will not in a bad state. Additionally, the result of two transactions depends on the transaction level of this system.

1. **You are a trusted advisor at a Bank and are told “We want to build a new application for managing transactions out of accounts (putting money into accounts and taking money out of accounts); we propose to use NoSQL DBs”. What would you suggest?**

Answer:

I would not recommend NoSQL DBs because it does not follow the consistency standard in ACID. However, for application managing transactions out of accounts, we need to make sure that the total value in two accounts should be same at the start and at the end of transaction. This requirement is the application of consistency principle in ACID so I think we should not use NoSQL DBs.

1. **You are building an application to handle Shopping Carts. What do you think would be acceptable trade-offs vis-a-vis CAP?**

Answer:

I think we need to consider availability and partition tolerance instead of considering consistency first. Here are the reasons.

In scenarios of shopping carts, system wants to capture as many choices as possible about what a user or customer is doing, but it isn’t critical that the database is constantly up to date because it does not include the payment process. The shopping carts simply just needs to be accessible and available even when network connections aren’t working. In such systems, only one machine can accept modifications while the reads can be done from all machines. The modifications flow from that one machine to the rest. Such systems are highly available as there are multiple machines to serve. Also, such systems are partition tolerant because if one machine goes down, there are other machines available to take up that responsibility. Since it takes time for the data to reach other machines from the node A, the other machine would be serving older data. This causes inconsistency. Though the data is eventually going to reach all machine and after a while, things are going to okay.

1. **You are building an application to handle Shopping Carts check out. What do you think would be acceptable trade-offs vis-a-vis CAP?**

Answer:

I think we need to consider consistency and partition tolerance instead of considering availability first. Here are the reasons.

High consistency means that all reads receive the most recent write or an error. For the shopping carts check, customers need to receive the recent feedback when they make payment to avoid paying again. Therefore, making sure the payment information (data in the database) is up to date even though the network is unreliable or fails is so important. We need to keep consistency. Plus, waiting for the system’s response of checking information (keep consistency) is acceptable so the availability is not required so much. In order words, considering the possible risk posted to the customer because of double payment, consistency comes before availability.