# M2 Case Study Analysis: Craigslist

*Please also review the grading rubric for this assignment in Canvas.*

**Answer the following questions after watching the case study video in Canvas**

**Question 1**

* Describe the architecture with the master and slave database environment cluster. What was it like before MongoDB and what was it like after?

Before MongoDB, there are huge master server and slave servers needed to implement big data as described in Craigslist case.

After MongoDB, it uses collection to store data. The MongoDB has a primary server to handle data and do the back-up copies of database instance to secondary server, which is named replica set. If we’re processing big data, we can shard it and range some indexed value to assign to different replica sets that can balance the data set with interacting with 3 config servers. The shard key for each shard can help users to save time to find specific values.

* + What type of NoSQL database was implemented and how much data was it intended to encompass?

Document-based data model;5 billion documents

**Question 2**

* What was the length of time it took to run the ALTER table query that the speaker referenced when Craigslist was operating in a relational environment?

It ran on Master server for 1 month and plugged in time used on slave servers with totally 2 months.

* + Why did it take so long?

Because when they wanted to change the schema type, the data in the Archive is too large to be changed as time goes by that it’ll take a long time.

**Question 3**

* Describe some of the hardware failures that Craigslist faced with their datacenter in the relational clustered environment

The way to look for the data in Archive is difficult because they have to insert data with indexes are haven’t used which will leads to a huge storage in the hardware so the hardware can’t deliver stuff successfully all the time, that is the hardware failures.

**Question 4**

* What were the lessons learned that Craigslist identified when moving from RDMS to MongoDB?
  + Explain each one in detail and provide your recommendation on how they might use today’s technology (e.g., AWS cloud environment) to assist in some of their lessons learned.

1. When the amount of data great exceed the RAM available, the operation can not deploy.
2. When they were testing the replica sets, there’re lots of reboots happened and without data reclone and DNS changes.
3. The encoding of the MongoDB combined old data and new data that are not all in UTF-8 format that will leads to fail. So keeping the encoding house in order.
4. Transforming the relational data to documents that data includes too many outliers are hassle to deal with in document types.
5. MongoDB can’t recognize the string and numerical data and some specific programming language don’t need to define the data type.

The aws cloud environment has a huge enough storage server in different regions to process big data that can avoid exceeding the RAM.

The aws can also automatically change a field data type from string to int.

The aws can detect outliers and use dedicated transforms to handle outliers in AWS Glue DataBrew.