Avi Skidelsky, DAV-6100, M2 Assignment

1) The way that the database was structured in Craigslist was structured in a relational format such that it was mirrored on a backend using what they referred to as the “archive” which, as described in the speech, is where listings go to die. The way that it was structured was such that a posting existed on the “live” schema and after a period of time it was migrated to the archive which was an exact copy of the live schema so that if a posting needed to referenced from the archive it could be done in the same manner that it would be in the live schema. Post MongoDB, the archive is organized into a document style NoSQL database so that each listing could live on its own and not as a relation. It was originally designed to be able to handle five billion documents.

2) It took about a month to run the ALTER table query on the archive. This is because of the size of the archive in addition to all of the smaller tables that go along with it in a relational database. The archive schema and data mirrors that of the live data but not only that, it records old records from potentially years back as opposed to the live source which deletes records periodically. Thus, the archive is ever growing and huge.

3) Like any hardware system, the physical hardware the Craigslist used crashed and broke occasionally and someone would need to out there to fix it. To correct this, someone would physically have to go to the datacenter to reboot the entire system and there’s no telling how long that would take because of the size of the relational tables. Additionally, since the archive is ever growing, they had to keep upgrading their hardware in order to maintain it.

4) There were a few lessons that they took from this experience:
   - Know your hardware: in the Craigslist production environment, every service was on a different server dedicated to that specific action so they were able to get a better handle on controlling everything. In the development environment though, Mongo didn’t cooperate with multiple tasks being run at the same time.
   - Replica sets rock: by creating replica sets of what was working until a restart was needed they were able to set up a sort of “jumping off point” where they could restart at that point, already having transferred a portion of the data without any problems. This would be even better done by using some of the AWS services that allow for temporary, small batches of the data to be saved temporarily.
   - Know your data very well: the data that they had wasn’t in a uniform format, specifically UTF-8 which Mongo runs on. This caused a lot of troubleshooting while importing and reformatting every time the data form wasn’t compliant.
   - Know your data size: Mongo had their own size constraints and even though the files in the Craigslist database typically didn’t violate those constraints, and most files won’t, there are outliers that need to be addressed and by knowing the constraints and applying that information to your data beforehand you can save steps of going back and reformatting.
   - Knowing your data type: storing isn’t the biggest issue when it comes to conflicting data types and misclassification, but querying is where that becomes an issue, causing issues searching for the wrong information.

- Know sharding: even though there are balancers in place to do this for you, it’s helpful and possible to know what should be split into which shards so that it’s in the exact format that the end user would find it useful.