http://www.mongodb.com/nosql-explained

https://www.digitalocean.com/community/tutorials/understanding-sql-and-nosql-databases-and-different-database-models

SQL vs NoSQL

* develeoped as a response to higher volumes of data
* Allows for more frequent access, faster processing

NoSQL

* Stores data as a key pair with the key pointing to a document
* Document can be key-value, key-array, or key-nested doucments
* more scalable and better performance
* take large volumes of structured, semi-structured and unstructured data
* quick iteration, frequent code pushes
* efficient architecture
* Dynamic schemas - allows you to store additional data instead of adding column to database. This requires you to update yoru entire database and can be time consuming with lots of downtime.
* NoSQL can allow the insertaion of data, no service interruptions, faster code integration.
* Structure
  + SQL requires defined attributes to hold the data
  + NoSQL allow free flow operations
* Querying
  + SQL queries are pretty standardized
  + NoSQL has unique ways to work for each database
* Scaling
  + Bot can scale vertically pretty easily but NoSQL makes it easier to scale horizontally. Cluster of machines
  + SQL traditionally one server to allow continuous access and relaiabliity. Scaling vertically means improve capacity of single server
  + Auto-sharding: Spread data across a number of srevers, without the application being aware of the server pool.
  + When server goes down it can be replaced with the application being disrupted
* Easier to get support for SQL because of how long they've been around

Summary

* Types- only sql database vs. many types like key-value, doucment database, column stores, adn graph database
* Development history- developed in 1970s vs. developed in 2000s to deal iwth limits of sql regarding scale,replication and usstructured dtaa
* Data storage models- records are kept storing rows in tables with columns storing a specifice piece of data about a record vs. key-values that only have two columns key and value. Or document database that nests values hierarchically.
* Schemas- strict structre with data types fixed in advanced. TO alter the datbase must be taken offline vs. adding new info on the fly dissilimar dta can be dstored together
* Scaling-Vertical meaning single server that is increasingly powerful vs Horiztonal meaning to add capacity to create a database across multiple servers