Period 3

1. What is NoSQL
    a. Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable
    b. characteristics apply such as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge amount of data and more
2. Why NoSQL
    a. a need to handle new, multi-structured data types or scale beyond the capacity constraints of existing systems
    b. desire to identify viable alternatives to expensive proprietary database software and hardware
    c. agility or speed of development, as companies look to adapt to the market more quickly and embrace agile development methodologies
    d. the relational data model is not well aligned with the needs of their applications. Consider:
        i. *Developers are working with applications that create new, rapidly changing data types — structured, semi-structured, unstructured and polymorphic data — and massive volumes of it*
        ii. *Applications that once served a finite audience are now delivered as services that must be always-on, accessible from many different devices and scaled globally*
        iii. *Organizations are now turning to scale-out architectures using open source software, commodity servers and cloud computing instead of large monolithic servers and storage infrastructure*
    e. When compared to relational databases, many NoSQL systems share several key characteristics including a more flexible data model, higher scalability, and superior performance
    f. Rather than spreading out a record across multiple columns and tables connected with foreign keys, each record and its associated (i.e., related) data are typically stored together in a single document. This simplifies data access and, in many cases, eliminates the need for expensive JOIN operations and complex, multi-record transactions
3. MongoDB and Redis classification
    a. MongoDB
        i. Document Model database
        ii. use a structure that is like JSON

iii. Documents provide an intuitive and natural way to model data that is closely aligned with object-oriented programming – each document is effectively an object
iv. each record and its associated (i.e., related) data are typically stored together in a single document. This simplifies data access and, in many cases, eliminates the need for expensive JOIN operations and complex, multi-record transactions
v. In a document database, the notion of a schema is dynamic: each document can contain different fields. This flexibility can be particularly helpful for modeling unstructured and polymorphic data. It also makes it easier to evolve an application during development, such as adding new fields
vi. Document databases are general purpose, useful for a wide variety of applications due to the flexibility of the data model, the ability to query on any field and the natural mapping of the document data model to objects in modern programming languages

b. Redis
   i. Key-Value Model
      1. Every item in the database is stored as an attribute name, or key, together with its value.
      2. Data can only be queried by the key. This model can be useful for representing polymorphic and unstructured data, as the database does not enforce a set schema across key-value pairs
      3. useful for a narrow set of applications that only query data by a single key value. The appeal of these systems is their performance and scalability, which can be highly optimized due to the simplicity of the data access patterns and opacity of the data itself

4.