**MongoDB Basic Research**

**Different between SQL and NoSQL database**

* SQL Database:
  + It is relational database.
  + It use structured query language and has a predefined schema.
  + It is vertically scalable.
  + It is table based.
  + It is better for multi-row transactions.
* NoSQL Database:
  + It is non-relational database.
  + It has hynamic schemas for unstructured data.
  + It is horizontally schalable.
  + It is document, key-value, graph or wide-column stores.
  + It is better for unstructured data like documents or JSON.

**Advantages and disadvantage of SQL**

* Advantages
  + **Faster Query Processing** – Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.
  + **No Coding Skills** – For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.
  + **Standardised Language** – Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.
  + **Portable** – It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.
  + **Interactive Language** – Easy to learn and understand, answers to complex queries can be received in seconds.
  + **Multiple data views**
* Disadvantages
  + **Complex Interface** – SQL has a difficult interface that makes few users uncomfortable while dealing with the database.
  + **Cost** – Some versions are costly and hence, programmers cannot access it.
  + **Partial Control** – Due to hidden business rules, complete control is not given to the database.

**Advantages and disadvantage of NoSQL**

* Advantages
  + It is table-less and easier to manage. It provides a higher level of flexibility with newer data models.
  + Lower cost: can be an appealing solution for smaller organizations with limited budgets.
  + It is optimized for specific data models that enable higher performance than trying to accomplish similar functionality with relational database.
* Disadvantages
  + Don’t have the reliability functions which Relational Database have (don’t support ACID).
  + Implement own code for support ACID in the system may reduce the number of safe applications that commit transactions, for example bank systems.
  + NoSQL are very new compared to Relational Databases, which means that are far less stable and may have a lot less functionalities.

**List NoSQL database available other than MongoDB**

* **Redis**: is an open source in-memory data structure server and NoSQL database.
* **Amazon DynamoDB**: is an Amazon Web Services.
* **Apache Cassandra**: is a free and open-source, distributed, wide column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.
* **Apache Hbase**: is an open source, distributed, versioned, non-relational database modeled after Google’s Bigtable.
* **Apache CouchDB**: is an HTTP + JSON document database with Map Reduce views and bi-directional replication.
* **RavenDB**: is fully transactional (ACID) across the database and throughout clusters.
* **Google Cloud Datastore**: is a NoSQL schemaless database as a service, supporting diverse data type. The database is managed; Google manages sharding and replication and prices according to storage and activity.
* **Apache Drill**: is a schema-free query engine for use with NoSQL or Hadoop or file storage systems and databases.

**JSON documents in MongoDB**

* In MongoDB, data is stored as documents. These documents are stored in MongoDB in JSON format.
* JSON documents support embedded fields, so related data and lists of data can be stored with the document instead of an external table.
* JSON is formatted as name/value pairs. In JSON documents, fieldnames and values are separated by a colon, fieldname and value pairs are separated by commas, and sets of fields are encapsulated in “curly braces” ({}).