**Model, implement and query a selected NoSQL database**

Before NoSQL came into picture, every organisation or business have used relational database which can be called as SQL database that consists of tables and relations. The structure of this is very rigid and scaling was not compatible with joins. It’s very difficult & expensive to handle SQL database while scaling them on powerful servers. To achieve proper scaling on SQL it has to be distributed on multiple servers. Poor time complexity is the other problem with SQL scaling, which is also a characteristic of joins, because a join operation is performed in SQL and database shows results by combining two or more tables.

NoSQL helps us in solving issues of SQL, as it follows horizontal scaling in which it splits up data into smaller segment and perform all queries within one segment. NoSQL uses hash table where value of each key is a b-tree. To spread data across an unlimited number of nodes, partition key is used. The sharding mechanism allows NoSQL to perform better without degradation on an infinite scale.

**NoSQL Models**

NoSQL data models does not need a relationship among data to establish design. Instead, it is simple due to the no need of schema, or a flexible schema which can be used or not used. As a result it is cost effective and cheaper in process. Not just that, it is highly effiecient in performing queries and giving output in seconds by analysing millions of data.

There are four types of NoSQL Models:

1. Key-value store

This is one of the most common data models, where each entry is made up of a key and a value. It is indeed the simplest of database which is built specifically for high-performance requirements. To retrieve data, instead of writing queries, keys are used.

## 2. Document based store

Their structure is much more similar json document, where pairs of fields and values are been stored. As a result wide variety of datatypes and data structures are supported by this type of nosql model. To retrieve data, queries are written by using powerful query languages to get field value types

## 3. Column based store

Data structure of this model is similar to sql model as it uses table but there is a difference. The tables in this model are flexible and scalable unlike the rigid structure of sql. There is a key present in rows which relates one or more columns, which creates column families. Since its similar to SQL, therefore queries are used to retrieve data.

## 4. Graph based store

For this task I am going to use MongoDB tool for modeling the key-value store database. First of all I downloaded the mongodb from <https://www.mongodb.com/try/download/community> and installed it. Then I created the connection by clicking on connect which can be seen from below picture.

Then I clicked on 'Import Data' to add 'clean.csv' to my database, and imported the data.

![image.png](attachment:image.png)

Then I performed a query { 'Location' : 'Rupert Streeet' } for station 'Rupert Street' which yielded 52584 results.

![image.png](attachment:image.png)

By double clicking on an document i was able to see the datatype where I can even update it

![image.png](attachment:image.png)

To know the details of schema I clicked on 'Schema' tab and analyzed the schema of my database.

![image.png](attachment:image.png)

To know the details of schema I clicked on 'Schema' tab and analyzed the schema of my database.

![image.png](attachment:image.png)

To know the details of schema I clicked on 'Schema' tab and analyzed the schema of my database.

![image.png](attachment:image.png)

Then I clicked on Explain plan which gave me full detail about the execution of query

![image.png](attachment:image.png)

By clicking on index i got detail about indexes

![image.png](attachment:image.png)