Key-Value NoSQL:

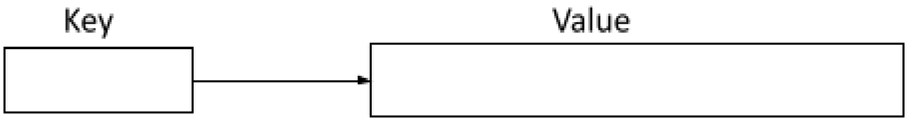
NoSQL has become quite popular lately, with some of the most popular platforms and services relying on it to deliver content to use with lightning speed. It has a variety of database types, but the most popular of them is Key-Value storage or we call it Key-Value NoSQL Database.

Its data model is extremely simple, which allows it to perform better and faster than a relational database. It also allows flexibility in storing and easy retrieval of data.

- **How does this work ?**

  It is quite simple to understand though. A value that is basically any piece of data or information is saved with a key to identify its location at time of operation. It's design concept is similar to that of an array or map in any programming language.
  Although it's stored in a continuous manner in the database system.

  It's representation is as shown below:



  Note: Now here value can be any piece of data in any format.

  They are considered so good because here information is stored as a sort of opaque blob (sizable piece of data), rather than discrete data.
  No index is required for the database to boost its performance. It is already well equipped within its structure to perform optimally.

  It depends on certains command to operate it like,
  Like,

  1. get(key): fetch us the value of the given key.
  2. put(key,value): Creates or updates a value of the given key.
  3. delete(key): Deletes the value for the given key.

4. execute(key): Invoke an operation to a value of the given key.

Although this may lack control of data, it gets overshadowed with the fact that this db is so fast and reliable.

- **Some Benefits of Key-Value NoSQL database:**

  It holds certain benefits with it like,

  - Highly Scalable Horizontally
  - Simpler queries
  - Mobility i.e. they are easy to move from one system to another without the need for new architecture or changing the code.
  - Handling concurrency issues better as only need to resolve one key.
  - Consistency.
  - Easy and fast retrieval of data.
  - Integrated caching

➢ Key-Value NoSQL Database - *Redis, Amazon DynamoDB, Oracle NoSQL, Aerospike*

- **When to use these kinds of databases?**

  Our Relational databases are not really compatible to handle a heavy number of read/write operations, which is where key-value db's shine.

  Since it's easily scalable, it can handle a high number of users at any given point.

  Also, with it's built-in redundancy feature, it can take care of lost storage or data without any difficulties.

  It is also preferable to be used when dealing with:

  1. Fulfills need to provide users with session management on quite a large scale.
  2. Can help in the formation of recommendation engines for products or anything.
  3. Can be used to advertise Advertisements to users depending on their current data usage.

  It is also preferred to be used for big data research or even for the seasonal surges in purchasing on platforms.

- **Limitations:**

- No relationships among Multiple-Data.
- Can’t Query Data by 'value'.
- Query is performed on one key at a time.

Therefore, the real deal with the Key-Value NoSQL database is that it’s simple. Although it might create issues when dealing with complex issues like financial transactions.

Its purpose is to bridge the gaps of relational databases.However,, by mixing both relational and non-relational together we can come up with a more efficient pipeline, being it for data analysis or managing users.