Data Modeling

Schema on read, storing raw data but applying schema as and when based on the purpose of reading, opens up vast possibilities dealing with futuristic requirements. This also sometimes refered as full fidelity of data. Same data set can be utilized by various teams for different applications. Hadoop provides datastore to accumulate big data with distributed storage.

Still it plays a vital role about making appropriate selection from available options i.e. File formats - sequential, record columnar, etc. Compression codecs - snappy and more. Metadata handling - relational, key-values like HBase, etc and the basic files and directories structures itself.

File Types include plain text file, sequential file, record columnar file and container format.

Sequential file is mostly used to combine very small files into a single file with markers. Markers indicates block boundaries. Markers help to split and allocate blocks appropriately. As hadoop optimized for big size datafiles, one of the example is having less number of files mean less metadata entries in memory to handle by namenode.

Serialization has to do with saving of object (data structures) to binary bytes and Deserialization is other way around. For this the protocol stores more than just data itself but also it needs to store metadata - details about members, class, parents, etc. to be able to deserialize correctly. The word count map reduce program uses Hadoop Writables. Any 'Writeable' is a serializable object which implements a simple, efficient, serialization protocol, based on DataInput and DataOutput. But there are containers like apache avro which allows to handle serialization.

Record columnar (RCFile) actually works to store column oriented data set physically. Particularly helpful for analytics over columns.

XML and JSON poses internal requirements of how to detect records and distribute. Container formats like avro enables storing such data too systematically.

Explore about Snapy, ORCFile, Parquet like tools to achieve more.

Source:
https://www.oreilly.com/library/view/hadoop-application-architectures/9781491910313/ch01.html