|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Heterogeneous Data * High performance * High scalability – Sharding * No Joins * Cloud distributions | * Not for relational data * Not as strongly ACID-compliant (Atomic, Consistency, Isolation, Durability) as the more well-established RDBMS systems. * Complex joining * Doesn’t support transactions * Can lead to duplication of data * Requires correct indexing for speed increases * High data consumption, low normalisation * Limited data size (16MB) |

**Use cases for MongoDB**

* Maintaining location-based data — Geospatial data.
* Real-time analytics and high-speed logging, caching, and high scalability.

**Pros ppl can pick from:**

* Document oriented
* High performance
* High availability — Replication
* High scalability – Sharding
* Dynamic — No rigid schema.
* Flexible – field addition/deletion have less or no impact on the application
* Heterogeneous Data
* No Joins
* Distributed
* Data Representation in JSON or BSON
* Geospatial support
* Easy Integration with BigData Hadoop
* Document-based query language that’s nearly as powerful as SQL
* Cloud distributions such as AWS, Microsoft, RedHat,dotCloud and SoftLayer etc:-. In fact, MongoDB is built for the cloud. Its native scale-out architecture, enabled by ‘sharding,’ aligns well with the horizontal scaling and agility afforded by cloud computing.

**Integrating large amounts of diverse data:** If you are bringing together tens or hundreds of data sources, the flexibility and power of the document model can create a unified single view in ways that other databases cannot. MongoDB has succeeded in bringing such projects to life when approaches using other databases failed.

**Describing complex data structures that evolve:** Document databases allow embedding of documents to describe nested structures and easily tolerate variations in data in generations of documents. Specialized data formats like geospatial are efficiently supported. This results in a repository that is resilient and doesn’t break or need to be redesigned every time something changes.

**Delivering data in high-performance applications:** MongoDB’s scale-out architecture can support huge numbers of transactions on humongous databases. Unlike other databases that either cannot support such scale or can only do so with massive amounts of engineering and additional components, MongoDB has a clear path to scalability because of the way it was designed. MongoDB is scalable out of the box.

**Supporting hybrid and multi-cloud applications:** MongoDB can be deployed and run on a desktop, a huge cluster of computers in a data center, or in a public cloud, either as installed software or through MongoDB Atlas, a database as a service product. If you have applications that need to run wherever they make sense, MongoDB supports any configuration now and in the future.

**Supporting agile development and collaboration:** Document databases put developers in charge of the data. Data becomes like code that is friendly to developers. This is far different from making developers use a strange system that requires a specialist. Document databases also allow evolution of the structure of the data as needs are better understood. Collaboration and governance can take place by allowing one team to control one part of a document and another team to control another part.