|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Efficiency** | **Scalability** | **Characteristics** | **Performance** |
| **Relational** | 1. Very Efficient with structured data | 1. Not much scalable as it is slower with large volume of data.  2. Vertically scalable | 1. It has a fixed Schema supported.  2. SQL as query language.  3. Transaction is supported (ACID). | They perform good with small data, but worse with large amount of data. |
| **Key - Value** | 1. Not as efficient as RDBMS when amount of data is small. Efficient than RDBMS when amount of data is large. | 1. Highly scalable. | 1. Schema free  2. SQL not supported  3. In-memory database  3. It supports transactions (ACID). | Performance depends on the hardware resources, as it is an in-memory database. More Hardware, more speed. |
| **Columnar** | 1. Efficient with semi structured and non-structured data. | 1. Highly scalable  2. It can handle TB to PB amount of data. | 1. Schema free  2. Column oriented  3. Transaction is not supported. | Fast with large amount of data. Can efficiently and effectively retrieve a large chunk of rows better than any RDBMS. |
| **Document** | 1. Efficient with non-structured data | 1. Highly scalable | 1.Schema free.  2. SQL not supported  3. Transaction not supported. | Very fast and even with large amount of data. Not much faster as compared to relational database when data is small. |
| **Graph** | 1. Efficient than RDBMS when there are huge number of joins and relationships. | 1. Highly scalable | 1. Graph based schema  2. Good for large no. of relations and joins.  3. Transactions are supported.(ACID) | Better performance than RDBMS. |