**Database Families**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Family | Examples | Data model | Scalability model | Use cases |
| Simple and Sophisticated key-value stores | Memcached (simple), Riak, Redis (Sophisticated) | Key-value, where the value is a binary blob. Sophisticated ones are persistent and may, for example, define data types for values, offer high vailability and fault-tolerance. | Variable. Memcached can scale across nodes, converting all available RAM into a single, monolithic datastore. | Example: Caching search results, session data |
| Wide-Column Stores | HBase, Cassandra | Variable. Cassandra uses a key-value structure known as a column. HBase stores binary blobs. | Eventually consistent, multinode distribution for high availability and easy failover. | High-throughput use cases - activity feeds, message queues, caching, Security analytics using network traffic and log data |
| Document DB | Mongo DB, Couch DB | Documents which encapsulate and encode data (or information) in some standard formats or encodings. Documents can have varying fields (schemaless) | Strong consistency, a rich data model, rich query syntax, and secondary indexes. Highly scalable – easy replication and sharding. | Rich query syntax makes it general-purpose like RDBMS. |
| RDBMS | Oracle Database, IBM DB2, Microsoft SQL Server, MySQL, PostgreSQL | Tables | Vertical scaling. Limited support for clustering and manual partitioning | System requiring transactions (banking, finance) or SQL. |

**Note**: Apart from these, there are other Database models like Graph and object databases.