Datastore Docstore Data warehouse

- store data in **relational tables** with a **fixed schema**. The data is **denormalized** (meaning data is pre-aggregated or flattened) for fast read access.
- usually sore large hostorical datasets that have gone through ETL processes, where data is structured and optimized for analysis.
- Designed for **complex analytical queries** that are **read-heavy** and typically take longer to execute (due to large datasets and aggregation). Data warehouses support **SQL-based analytics** on structured, historical data.
- Data is **not updated in real-time** but is usually refreshed periodically (e.g., daily, weekly) via **ETL** pipelines.
- typically **read-optimized** and don't provide full **ACID transactional** guarantees. Instead, they support **batch processing** and **ETL workflows**, which are designed for large-scale data analysis rather than real-time transaction integrity.

Summary of Key Differences

Feature	Datastore	Docstore	Data Warehouse
Purpose	Generic data storage (OLTP or OLAP)	Transactional database for semi-structured data	Analytical database optimized for large-scale queries
Data Model	Key-value, document, column-family, relational, graph	Document-based (JSON/BSON) and multi-model	Relational (denormalized for performance)
Data Type	Structured, semi-structured, unstructured	Semi-structured (JSON-like)	Structured, historical data
Consistency	Can be eventually consistent, strong consistency may be supported	ACID transactional guarantees	Typically eventual consistency, batch updates
Querying	Fast read and write operations (CRUD)	Real-time queries with flexible structure	Complex aggregations, OLAP queries
Primary Use Case	Real-time apps, caching, sessions, etc.	Real-time data with ACID support, high availability	Business intelligence, reporting, data analysis

Feature	Datastore	Docstore	Data Warehouse
Scalability	High scalability (horizontal scaling)	Horizontal scalability	Horizontal scaling, optimized for large data
Open-Source Examples	Redis, Cassandra, MongoDB, etc.	CouchDB, MongoDB, ArangoDB, etc.	Apache Hive, ClickHouse, Druid, Greenplum

---

Key Takeaways:

1. **Datastore**: A broad term for any system that stores data. It can be any type of database (relational, NoSQL, graph, etc.), and is generally used for **real-time transactional data**.
2. **Docstore**: A more specialized **multi-model transactional database** that stores **document-like data** (JSON) and supports **ACID transactions**. It's used for applications that need real-time data storage and high availability.
3. **Data Warehouse**: A **read-optimized** database designed for **analytical queries** over **historical** data, optimized for **large-scale aggregations** and business intelligence.