

Feature / Database	Azure Cosmos DB (Core SQL API)	AWS DynamoDB	AWS DocumentDB (MongoDB Compatible)	GCP Firestore	GCP Spanner
Database Type	Multi-model (Document, Key-Value, Graph) - primarily Document for SQL API	Key-Value / Document	Document (MongoDB API compatible)	Document	Relational with horizontal scalability, NewSQL
Data Model	Schemaless JSON documents	Schemaless Key-Value, JSON attributes	Schemaless BSON (MongoDB) documents	Schemaless JSON documents (Collections & Documents)	Relational (tables, rows, columns)
Query Language	SQL-like (Cosmos DB SQL)	Proprietary API (Key Condition & Filter Expressions)	MongoDB Query Language (MQL)	Firestore Query Language (GQL) - limited SQL-like	Standard SQL (ANSI 2011)
Consistency	Tunable (Strong, Bounded Staleness, Session, Consistent Prefix, Eventual)	Eventual (default), Strongly Consistent Reads (for primary key/LSI)	Eventual (default), Tunable for reads (read preference)	Strong (within transactions), Eventual for distributed queries	Strong (global ACID transactions)
Scalability	Global distribution, automatic partitioning, elastic scale	Serverless, automatic partitioning, massive scale	Managed, scalable MongoDB compatible clusters	Serverless, scales to large datasets, real-time sync	Globally distributed, highly scalable, relational
Indexing	Automatic indexing by default (all properties), configurable	Primary key (partition/sort), Secondary Indexes (LSI, GSI) - manual setup	Automatic for _id, manual for other fields	Automatic indexing (requires indexes for all queries)	Automatic indexing
Reactive Support (Java)	Strong (Azure SDK for Java)	AWS SDK for Java 2.x (async, non-blocking)	AWS SDK for Java (for MongoDB compatible driver)	Google Cloud Java Client Library (async, non-blocking)	Google Cloud Java Client Library (async, non-blocking)
Pricing Model	Request Units (RUs) per second, storage	Read/Write Capacity Units (RCUs/WCUs) or On-Demand, storage	Instances (EC2 types), storage	Document reads/writes, storage, network egress	Compute capacity, storage, network egress
Transactionality	ACID within a single logical partition (document or stored procedure), multi-doc transactions with batch	ACID within a single item or across multiple items in a single partition (transactional APIs)	ACID for single document, multi-document via transactions (limited across shards)	ACID for multi-document operations	Strong ACID global transactions
Use Cases	Web/mobile, IoT, retail, gaming, global-scale apps needing flexible schema and low latency	High-traffic web/mobile, gaming, ad-tech, IoT, real-time bidding, key-value workloads	MongoDB compatibility for existing apps, document-centric workloads	Mobile/web apps, real-time collaboration, user profiles, social apps	Mission-critical, global consistency, financial services, inventory