Google Cloud Datastore

- Google Cloud Datastore (Cloud Datastore) is a highly scalable, fully managed NoSQL database service offered by Google on the Google Cloud Platform.
- Cloud Datastore is meant for applications that demand reliability upon the highly available structured data at a fixed scale.
- Cloud Datastore—a document database built for automatic scaling, high performance, and ease of use.
- Datastore is Serverless . App Engine + Datastore automatically encrypted data.
- Cloud Datastore is built upon Google's Bigtable and Megastore technology. Datastore automatically handles **sharding** (splitting and distributing one logical data set across multiple databases that can be deployed across multiple servers) and replication, providing you with a highly available and durable database that scales automatically to handle your applications' load.
- Google Cloud Datastore offers high-end capabilities that include SQL-like queries, ACID transactions, indexes, and others, to help enhance the end outcomes.

DATASTORE	RDBMS
KIND	TABLE
TABLE	ROW
PROPERTY	COLUMN
KEY	RIMARY KEY

- Google Cloud Datastore allows the user to create databases either in Native or Datastore Mode.
    - **Native Mode** is designed for mobile and web apps
    - **Datastore Mode** is designed for new server projects.
- Similar to SQL Like Queries, GQL and supports multiple client libraries
- Multiple indexes. Data replication across different region. Fully managed with no downtime
- Export data from gcloud utility only.

- Cloud Datastore is best suitable for storing the transactions and hierarchical data.
    - ACID (Atomicity, Consistency, Isolation, and Durability) compliant and multi-document transactions are supported with Cloud Datastore.
    - Cloud Datastore has the potential to help support the primary, secondary & composite indexes.
    - Datastore by Google Cloud encrypts all of the data automatically before it can be written over the disk. And it is also offering Identity & Access Management (IAM).
- Cloud Datastore is good for product catalogs, user profiles and transactions that provide real time inventory and product details for a retailer. Datastore is not suitable for analytic data.
- Google Cloud Datastore offers two redundancy levels that depend upon the replications within multiple locations.
- The levels include Regional replication and multi-region replication.
- **Regional Replication**
    - Under regional replication, the data undergoes replication within at least 3 varying zones but within that same region.
    - Hence, this will make that database more resilient towards zonal outages.
    - Regional replication is preferable for implementing low write latency.
- **Multi Regional replication**
    - The multi-region replication allows replication of data, within multiple zones, across a minimum of two regions.
    - Hence, this brings out the result of enhanced availability and redundancy.

Features Of Cloud Datastore

Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. Datastore features include:

- **Atomic transactions -** Datastore can execute a set of operations where either all succeed, or none occur.
- **High availability of reads and writes -** Datastore runs in Google data centers, which use redundancy to minimize impact from points of failure.
- **Massive scalability with high performance -** Datastore uses a distributed architecture to automatically manage scaling. Datastore uses a mix of indexes and query constraints so your queries scale with the size of your result set, not the size of your data set.
- **Flexible storage and querying of data -** Datastore maps naturally to object-oriented and scripting languages, and is exposed to applications through multiple clients. It also provides a SQL-like query language.
- **Balance of strong and eventual consistency -** Datastore ensures that entity lookups by key and ancestor queries always receive strongly consistent data.
    - This storage service provides terabytes of capacity with a maximum unit size of one megabyte per entity.
- **Encryption at rest -** Datastore automatically encrypts all data before it is written to disk and automatically decrypts the data when read by an authorized user.
- **Fully managed with no planned downtime -** Google handles the administration of the Datastore service so you can focus on your application. Your application can still use Datastore when the service receives a planned upgrade.

Cloud Datastore pricing

- Cloud Datastore is the best for semi-structured application data that is used in app engines' applications.
- The pricing of Cloud Datastore is on entity
  - writes
  - reads
  - deletes
- Below breakdown is for US(Multi-Region)
- https://cloud.google.com/blog/products/gcp/google-cloud-datastore-simplifies-pricing-cuts-cost-dramatically-for-most-use-cases/

US (multi-region)

	FREE limit per day	PRICE above free limit (per unit)	Price Unit
Stored data	1 GB storage	$ 0.18	GB/Month
Entity Reads	50,000	$ 0.06	per 100,000 entities
Entity Writes	20,000	$ 0.18	per 100,000 entities
Entity Deletes	20,000	$ 0.02	per 100,000 entities
Small Operations	50,000	Free	-

Use cases of Cloud Datastore

- Datastore is ideal for applications that rely on highly available semi-structured data at scale. You can use Datastore to store and query all of the following types of data:
  - Product catalogs that provide real-time inventory and product details for a retailer.
  - User profiles that deliver a customized experience based on the user's past activities and preferences.
  - Transactions based on ACID properties, for example, transferring funds from one bank account to another.

Best Practices for Cloud Datastore

Here are a few best practices that can help you work with Cloud Datastore more effectively:

- **Use batch operations**—these are more efficient because they use the same overhead as one operation.
- **Roll back failed transactions**—if there is another request for the same resources, this will improve the latency of the retry operation.
- **Use asynchronous calls**—like in Firestore, prefer to use asynchronous calls if there is no data dependency of the result of a query.