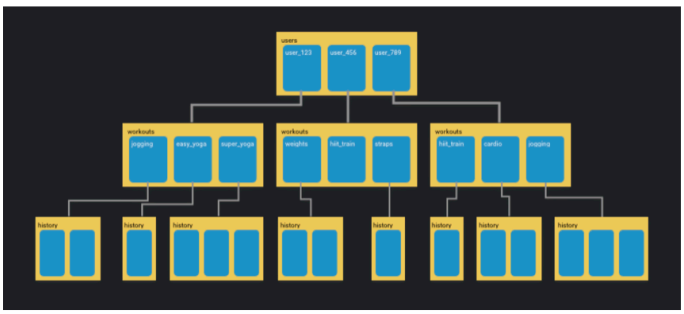
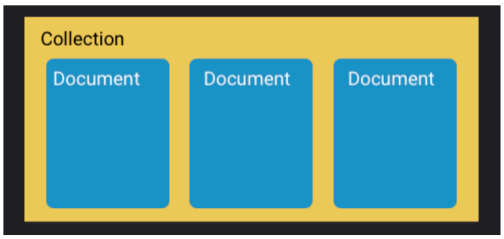
Google Cloud Firestore

- Cloud Firestore is the newest version of Datastore. Existing Datastore users can access these improvements by creating a new Firestore mode in database instance.
- In the future, all existing Datastore databases will be automatically upgraded to Firestore in Datastore mode.
- Cloud Firestore - a document-oriented, a NoSQL database storing key-value pairs and also a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud.
- Cloud Firestore leverages the technologies of Firebase technologies with the Google Cloud Platform. It performs real-time synchronization of data between client applications using event listeners.
- Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity.
- Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.
- Cloud Firestore is optimized for storing large collections of small documents. Documents may contain sub collections and nested objects, including strings, complex objects, lists, or other primitive fields.

FIRESTORE	RDBMS
COLLECTION GROUP	TABLE
DOCUMENT	ROW
FIELD	COLUMN
DOCUMENT ID	RIMARY KEY

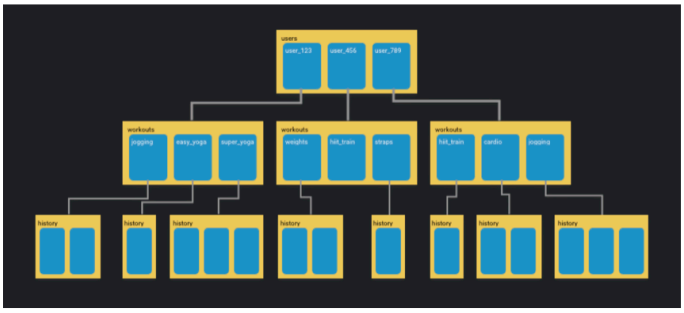


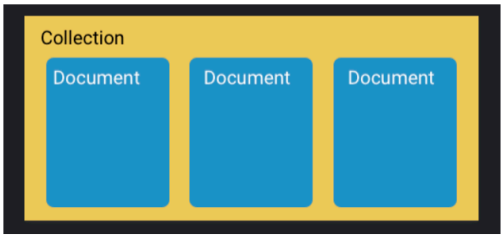


- Firestore creates these documents and collections implicitly. That is, when you assign data to a document or collection, Firestone creates the document or collection if it does not exist.
- Cloud Firestore supports Python, Node.js, Unity, Java, C++, REST, Go's native SDK, and RPC APIs.
- Cloud Firestore queries are highly expressive and flexible, letting use flat queries to fetch document level data without fetching the entire collection or nested sub collections.
- It also add real-time listeners to your application, to take a snapshot when data in the client application changes, and retrieve only the new changes rather than the entire database.
- Firestore is the next generation of Datastore. It adds
    - A new, strongly consistent storage layer - Flexibility .
    - A collection and document data model - Express querying.
    - Real time updates
    - Offline support
    - Designed to scale
    - Highly scalable NoSQL database
    - Collection & Document Model
    - User cannot use both native and datastore mode in the same project.

Features of Cloud Firestore

- Flexible data model—supports hierarchical data structures. Lets you store data in documents, which you can further organize into collections and sub-collections.





- Documents can support complex nested objects.
  - **Complex queries -** enable you to retrieve a specific single document or all documents in a collection, as long as it matches the query parameters.
    - Additionally, you can combine sorting and filtering, as well as include multiple filters in one query.
  - **Data sync -** updates data automatically on each connected device. The sync operation is designed to efficiently run simple one-time queries.
  - **Caching -** caches application data, allowing your application to write, read, monitor, and perform data query offline.
    - This mechanism syncs local changes with Cloud Firestore once the system comes back online.
  - **Automatic scaling -** Firestore scales data storage automatically, retaining the same query performance regardless of database size.
  - **Serverless development -** networking and authentication are handled using client side SDKs, with less need to coding.
  - **Backend security rules -** enabling complex validation rules on data.

- **Offline support -** databases can be accessed from user devices while offline on web browsers, iOS and Android.
- **Datastore mode -** support for the Cloud Datastore API, enabling applications that currently work with Google Cloud Datastore to switch to Firestore without code changes.

Cloud Firestore pricing

When you use Firestore, you are charged for the following:

- The **number of documents** you **read**, **write**, and **delete**.
- The **number of index entries** matched by aggregation queries - such as `count()`.
- The **amount of storage** that your database uses, including overhead for metadata and indexes.
- The **amount of network bandwidth** that you use.

All charges accrue daily.

Best Practices for Cloud Firestore

Here are a few best practices that will make the most of Cloud Firestore

- Database Location

Select a database location closest to your users, to reduce latency. You can select two types of locations:

- **Multi-regional location**—for improved availability, deploys the database in at least two Google Cloud regions.
- **Regional location**—provides lower cost and better write latency (because there is no need to synchronize with another region)

- Gradually Increasing Traffic

- Give Cloud Firestore time to prepare documents for higher traffic.
- To do this, gradually increase traffic to new collections or documents that are close in terms of lexicographic order.

- Start with up to 500 operations per second for a new collection, then increase traffic by 50% every five minutes. This is called the "500/50/5" rule. Make sure work is distributed evenly along the range of values of the document key.

- Indexes

- Minimize the number of indexes—too many indexes can increase write latency and storage costs.

- Optimizing Read and Write Operations

- Avoid writing to the same document several times per second.
- If possible, use asynchronous calls instead of synchronous calls, to minimize latency impact.

Comparison Firebase vs Firestore

- A brief comparison of Cloud Firestore vs Firebase.

Firebase vs. Firestore Comparison

	Firebase	Cloud Firestore
Summary	App development platform	NoSQL Scalable Database
Core Features	Databases, Cloud Functions, Storage, Analytics, A/B Testing, Authentication, etc.	Scalable hosting, multi region deployment, data synchronization.
Databases	Two. Firestore and The Real Time Database	Firestore
Database Type	Both are NoSQL	NoSQL
Plans	Spark and Blaze	Spark and Blaze
Free Tier	Yes	Yes
Pricing Model	Pay as you go	Pay as you go
Pricing	Depends on the service	Function of network out, database size, writes, reads, and deletes.