**Cloud Firestore & Datastore:**

**Cloud Datastore:**

* Google Cloud Datastore (Cloud Datastore) is a highly scalable, fully managed [NoSQL](https://en.wikipedia.org/wiki/NoSQL) database service offered by [Google](https://en.wikipedia.org/wiki/Google) on the [Google Cloud Platform](https://en.wikipedia.org/wiki/Google_Cloud_Platform).
* It follows atomic transaction, operations where all succeed, or none occur.
* Cloud Datastore is built upon Google's [Bigtable](https://en.wikipedia.org/wiki/Bigtable" \o "Bigtable) and Megastore technology.
* 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.
* Serverless
* **Document** kind data storage
* App Engine + Datastore automatically encrypted data
* Similar to SQL Like Queries, GQL and supports multiple client libraries
* Multiple indexes
* Data replication across different region
* Export data from gcloud utility only
* Fully managed with no downtime
* Good for
  + Product catalogs that provide real time inventory and product details for a retailer.
* [**https://cloud.google.com/datastore/docs/concepts/entities?\_ga=2.128048074.-779187871.1646201652**](https://cloud.google.com/datastore/docs/concepts/entities?_ga=2.128048074.-779187871.1646201652)

**Cloud Firestore:**

* Firestore is a NoSQL document database built for automatic scaling, high performance, and ease of application development.
  + While the Firestore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects.
* Firestore is the next generation of Datastore. It adds
  + A new, strongly consistent storage layer.
  + A collection and document data model.
  + Real time updates
* Highly scalable NoSQL database
* Collection & Document Model
* Two modes
  + native Mode
  + datastore mode
    - Firestore in datastore mode uses datastore system behavior but access firestore’s storage layer.
    - Removes limitations like
      * More consistent queries
      * Transactions are unlimited to entity groups, previously with datastore there was a limitation to 25 entity group. (Entity is a table)
      * Writes to an entity group are no longer limited to 1 per second.
* User cannot use both native and datastore mode in the same project
* Real-time updates
* Mobile and Web client libraries

**Entities:**

* Data objects in Firestore in Datastore mode are entities.
* Entity has one or more named properties, each of which can have one or more values.
* Datastore mode supports below datatypes as properties values:
  + Integers
  + Floating point numbers
  + Strings
  + Dates
  + Binary data
* Application can use firestore in datastore mode API to create, retrieve, update and delete entries.
* Each entity in a Datastore mode database has a key that uniquely identifies it.
* Key components:
  + Namespace of entity, which allow **multitenancy**
    - Firestore in Datastore mode allows a multitenant application to use separated silos of data for each tenant while still using: a single project. a single logical structure for the kinds. a single set of index definitions, because the kinds are the same logically for each tenant.
  + Kind of entity, which categorizes it
  + An identifier for the individual entity, either key Name String, an integer numeric id.
* An application can fetch an individual entity from the database using the entity’s key.

|  |  |  |
| --- | --- | --- |
| **Property** | **Datastore** | **Firestore** |
| **Data model** | Document database organized into documents and collections. | Entities organized into kinds and entity groups. |
| **Queries and transactions** | • Strongly consistent queries across the entire database  • Up to 500 documents per transaction across any number of collections.  • Limitation: No projection queries | * Removes the previous consistency limitations of Datastore   • Strongly consistent queries across the entire database  • Transactions can access any number of entity groups |
| Datastore v1 API support | No, requests are denied | Yes |
| Firestore v1 API support | Yes | No, requests are denied |
| Offline data persistence | The mobile and web client libraries support offline data persistence | Not supported |
| Client libraries | Firestore client libraries:  • Java • Python • PHP • Go • Ruby • C# • Node.js • Android • iOS • Web | Datastore Client libraries:  • Java • Python • PHP • Go • Ruby • C# • Node.js |
| Security | Identity and Access Management (IAM) manages database access  • Firestore Security Rules support serverless authentication and authorization for the mobile and web client libraries | IAM manages database access |
| Performance | Automatically scales to millions of concurrent clients. Max 10,000 writes per second. | Automatically scales to millions of writes per second |
| Console | Firebase Console and Cloud Console Firestore Viewer | Cloud Console Datastore Viewer |
| App Engine client library integration | Not supported in the App Engine standard environment Python 2.7 and PHP 5.5 runtimes Supported in the App Engine standard environment Python 3.7, PHP 7.2, Java 8, Go, and Node.js runtimes Supported in the App Engine flexible environment, all runtimes | Supported in all runtimes |

**Hands-on In Cloud Datastore and Firestore:**

**Resources:**

[**https://cloud.google.com/firestore/docs/quickstarts**](https://cloud.google.com/firestore/docs/quickstarts)

[**https://cloud.google.com/datastore/docs/store-query-data**](https://cloud.google.com/datastore/docs/store-query-data)