

Datastore Overview

Firestore in Datastore mode (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 dataset.
- **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](/datastore/docs/apis/gql/gql_reference) (/datastore/docs/apis/gql/gql_reference).
- **Strong consistency.** Datastore ensures that all queries are strongly consistent.
- **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. For more information, see [Server-Side Encryption](/datastore/docs/concepts/encryption-at-rest) (/datastore/docs/concepts/encryption-at-rest).
- **Fully managed with no planned downtime.** Google handles the administration of the service so you can focus on your application. Your application can still use Datastore when the service receives a planned upgrade.

Comparison with relational databases

While the Datastore interface has many of the same features similar to relational databases, as a NoSQL database, it varies in how it describes the relationships between data objects. Here's a high-level comparison of Datastore and relational database concepts:

Concept	Datastore	Firestore	Relational database
Category of object	Kind	Collection group	Table
One object	Entity	Document	Row

Concept	Datastore	Firestore	Relational database
Individual data for an object	Property	Field	Column
Unique ID for an object	Key	Document ID	Primary key

Unlike rows in a relational database table, Datastore entities of the same kind can have different properties, and different entities can have properties with the same name but different value types. These unique characteristics imply a different way of designing and managing data to take advantage of the ability to scale automatically. In particular, Datastore differs from a traditional relational database in the following important ways:

- Datastore is designed to automatically scale to very large data sets, allowing applications to maintain high performance as they receive more traffic:
 - Datastore writes scale by automatically distributing data as necessary.
 - Datastore reads scale because the only queries supported are those whose performance scales with the size of the result set (as opposed to the data set). This means that a query whose result set contains 100 entities performs the same whether it searches over a hundred entities or a million. This property is the key reason some types of queries are not supported.
- Because all queries are served by previously built indexes, the types of queries that can be executed are more restrictive than those allowed on a relational database with SQL. In particular, Datastore does not include support for join operations, inequality filtering on multiple properties, or filtering on data based on results of a subquery.
- Unlike relational databases which enforce a schema, Datastore is schemaless. It doesn't require entities of the same kind to have a consistent set of properties (although you can choose to enforce such a requirement in your own application code).

What it's good for

Datastore is ideal for applications that rely on highly available 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](https://en.wikipedia.org/wiki/ACID) (<https://en.wikipedia.org/wiki/ACID>) properties. For example, transferring funds from one bank account to another.

Other storage and database options

Datastore is not ideal for every use case. For example, Datastore is not a relational database, and it is not an effective solution for analytic data.

Here are some common scenarios where you should probably consider an alternative to Datastore:

- If you need a relational database with full SQL support for an online transaction processing (OLTP) system, consider [Cloud SQL](#) (/sql).
- If you don't require support for ACID transactions or if your data is not highly structured, consider [Bigtable](#) (/bigtable).
- If you need interactive querying in an online analytical processing (OLAP) system, consider [BigQuery](#) (/bigquery).
- If you need to store large immutable blobs, such as large images or movies, consider [Cloud Storage](#) (/storage).

For more information about other database options, see the [overview of database services](#) (/products/databases).

What's next

- [Learn how to store and query data using the Google Cloud console](#) (/datastore/docs/store-query-data)
- [Learn about the Datastore data model](#) (/datastore/docs/concepts/entities)
- [View best practices for Datastore](#) (/datastore/docs/best-practices)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-03-27 UTC.