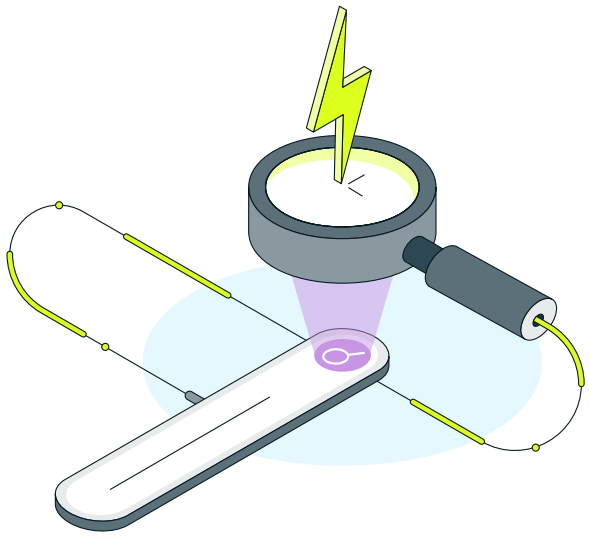
DATASHEET

Redis Query Engine

Run powerful data queries & search in real time

Use for free on Redis Cloud →



Introducing Redis Query Engine

Redis is a real-time key-value datastore known for its speed, flexibility, and simplicity.

Traditionally, data values are stored and retrieved from Redis by unique key names. This is great when you're working with single key-value pairs or with small number of pairs in sequence. But when you're trying to work with many key-value pairs, especially simultaneously or in groupings selected in real time, you need a more sophisticated option.

We built Redis Query Engine from the ground up as a native Redis add-on that breaks through the limitations imposed by exact key matching. You no longer need to handle data processing yourself or go outside of Redis.

The Redis Query Engine introduces a query language and engine that lets you run powerful data queries and searches in real time, directly on Redis.

Key product features

Quickly **query** data with high precision.

Run powerful, low latency **searches.**

Surgically **aggregate** data in real time.

Easily define high performance **indexes.**

What customers say

"Redis maintains **sub-5 ms response times for inventory queries**–despite high traffic–improving reliability and speed during critical shopping periods."

Loblaw COMPANIES LIMITED

"Redis vector similarity search (VSS) helped Asurion's AI assistant achieve **sub-10 ms response times, a 50% improvement.**"

asurion

"It was incredibly faster than any other solution, including our internal implementation. We saw **vector search speeds go from 2 seconds to 10 milliseconds.**"

Relevance AI

Query

Queries (i.e., data queries) are the primary way to retrieve Redis data by matching to tags or attributes. Redis Query Engine supports concurrent, low-latency insertion and updates of documents.

Key query capabilities for Redis Query Engine include multi-field queries, query on JSON documents, boolean queries with AND, OR, and NOT operators between subqueries, optional query clauses, numeric filters and ranges, and geo-filtering using Redis geo commands.

2 query commands are available: FT.SEARCH and FT.AGGREGATE.

1. FT.SEARCH to perform selections (i.e., return all documents that fulfill specific criteria) and projections (i.e., return specific fields of the result set or map to calculated field values) only. Learn more about exact match queries, range queries, geospatial queries, combined queries.
2. Use FT.AGGREGATE command to apply mapping functions, group, or aggregate data to collect and summarize data across several fields. Learn more about aggregation queries.

Running a query on the 'index:product' index to return a list of all those that have the value 'redis' in the 'brand' field is very simple:

```
FT.SEARCH index:product "@brand:{redis}"
```

Search

Search is often used to get the most similar data in Redis by best semantic match or keyword match. Full-text search, geospatial search, and vector search over vector embeddings are subtypes of search.

Redis Query Engine has search built in natively with a powerful autocomplete engine that can do fuzzy matching and with multiple scoring models and sorting by values. Concurrent searches can be run on Redis Query Engine and allows long-running searches without blocking Redis.

Key full-text search capabilities for Redis Query Engine include: prefix-based searches, exact phrase search and slop-based search, field weights, auto-complete and fuzzy prefix suggestions, and stemming-based query expansion for many languages using Snowball.

It also supports custom functions for query expansion and scoring (see extensions) and document ranking.

Key geospatial search capabilities for Redis Query Engine include search for locations within a specific radius, search based on geometric shapes, such as polygons (representing building layout, lake, etc.), and search on points within a geometric shape.

Key vector search capabilities for Redis Query Engine include running vector search queries with the FT.SEARCH or FT.AGGREGATE commands, semantic searches over vector embeddings, augmented searches with filtering over text/numerical/geospatial/tag metadata, and advanced querying strategies with vector fields including k-nearest neighbor (KNN), vector range queries, and metadata filters.

Running a search on the 'index:product' index to return a list of all those that mention words similar to 'RAM & SSD' such as 'ram and ssd' is straightforward.

```
FT.SEARCH index:product "RAM & SSD"
```

Use Redis Query Engine for free on Redis Cloud. Get started →