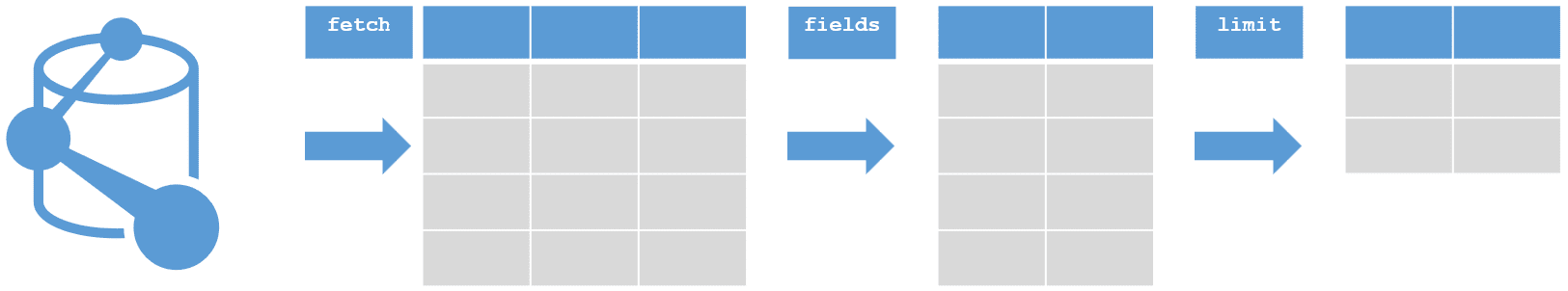
**Dynatrace Query Language**

Dynatrace Query Language (DQL) is a powerful tool to explore your data and discover patterns, identify anomalies and outliers, create statistical modeling, and more based on data stored in [Dynatrace Grail storage](https://docs.dynatrace.com/docs/discover-dynatrace/platform/grail/dynatrace-grail).



**Load Data:**

The DQL fetch command defines which data to load from Dynatrace and optionally process in the following processing pipeline steps.

fetch logs

| filter loglevel == "ERROR"

| summarize numErr = count()

**It contains three commands—fetch, filter, and summarize—each separated by a pipe.**

You can also change the data type to events by using the fetch command and following it with events.

fetch events

| summarize Total\_amount = sum(amount)

**Specify timeframe**

fetch logs, from:now() - 2h

This example queries logs from the last 24 hours, excluding the last two hours:

fetch logs, from:now() - 24h, to:now() - 2h

You can also use absolute time ranges with the timeframe parameter:

fetch logs, timeframe:"2021-10-20T00:00:00Z/2021-10-28T12:00:00Z"

**Filter**

fetch logs, from:now() - 2h

| filter loglevel == "SEVERE" or loglevel == "ERROR"

**Select a subset of fields**

fetch logs

| fields timestamp, loglevel, log.source, content

**Order results with sort**

By default, the sort command sorts records in ascending order. In the following example, we sort results in descending order.

fetch logs

| filter loglevel == "SEVERE" or loglevel == "ERROR"

| fields timestamp, loglevel, dt.process.name, host.name, content

| limit 5

| sort timestamp desc

**Aggregations**

**Aggregations over time**

fetch logs

| filter loglevel == "SEVERE" or loglevel == "ERROR"

| makeTimeseries count = count(), by:loglevel, interval:5m