

Eager mode

We understand the challenges that come with handling and analysing large volumes of data. That's why we created the eager mode. A lazy execution mode for optimising your data analysis workflow.

Why eager mode

Leveraging lazy execution can significantly enhance your data analysis workflow. By deferring computations until necessary, it enables you to build complex data transformation pipelines without incurring the performance penalty of executing each operation immediately. This approach not only minimises memory usage by avoiding the creation of intermediate data structures but also allows us to optimise the entire computation graph, selecting the most efficient execution strategy. Whether you're dealing with large datasets or need to streamline your data processing tasks, Eager mode ensures that your operations are both fast and resource-efficient, making it a powerful tool in your data analysis arsenal.

How to use it

```
...  
  
Copy loader=DatasetsLoader() df=loader.load("tokens-daily-prices-mcap-volume", eager=True)  
  
...  
  
...  
  
Copy df_filtered=df.drop("market_cap").filter(token="WETH").limit(3) df_filtered.collect()  
  
...
```

After using the `collect()` method, the result is loaded into memory. Before executing the collect method, you can add as many operations as you want. Here is the result of the above code snippet:

?

[Previous DatasetsLoader](#) [Next Cache management](#)

Last updated 1 day ago