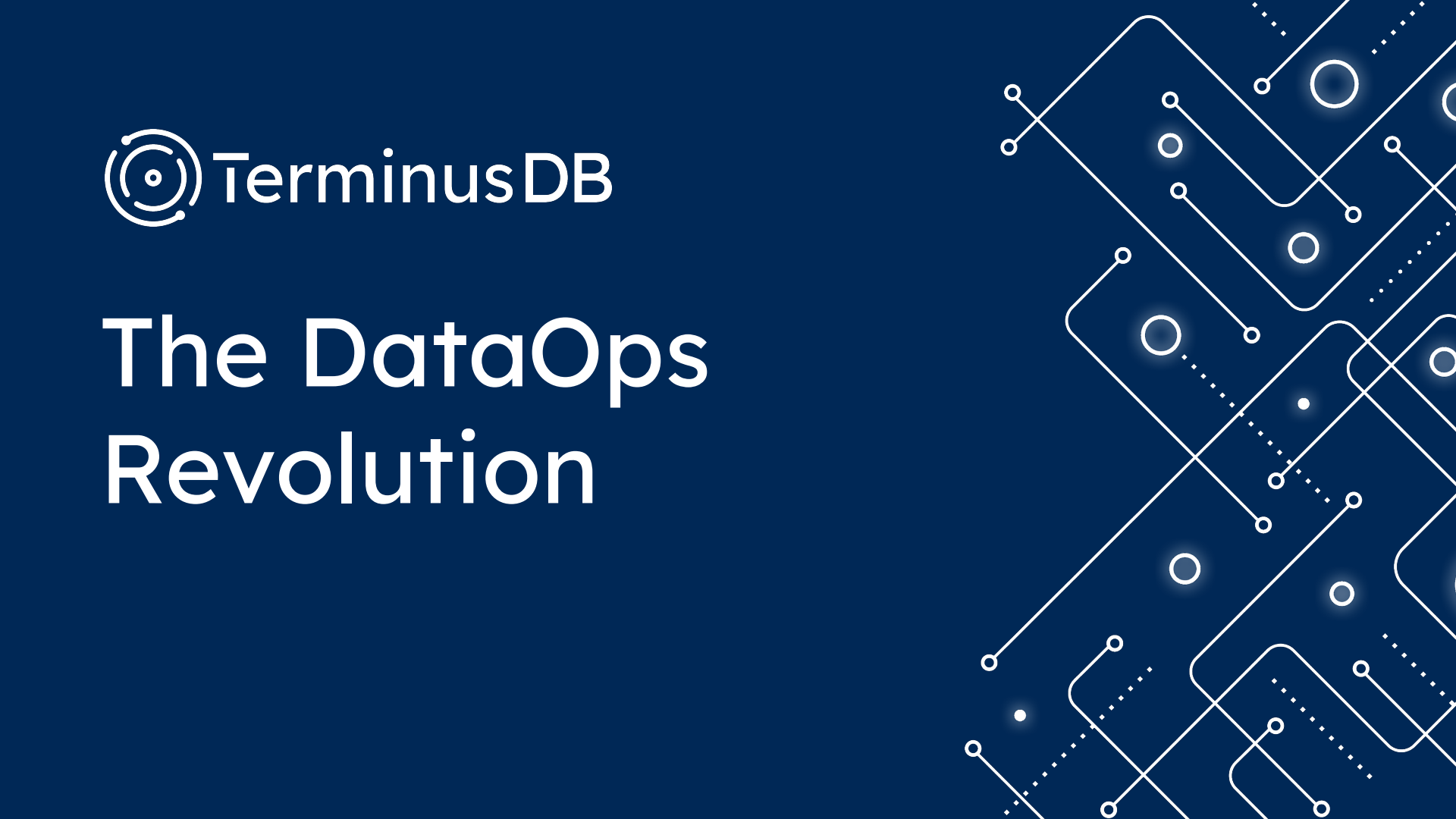
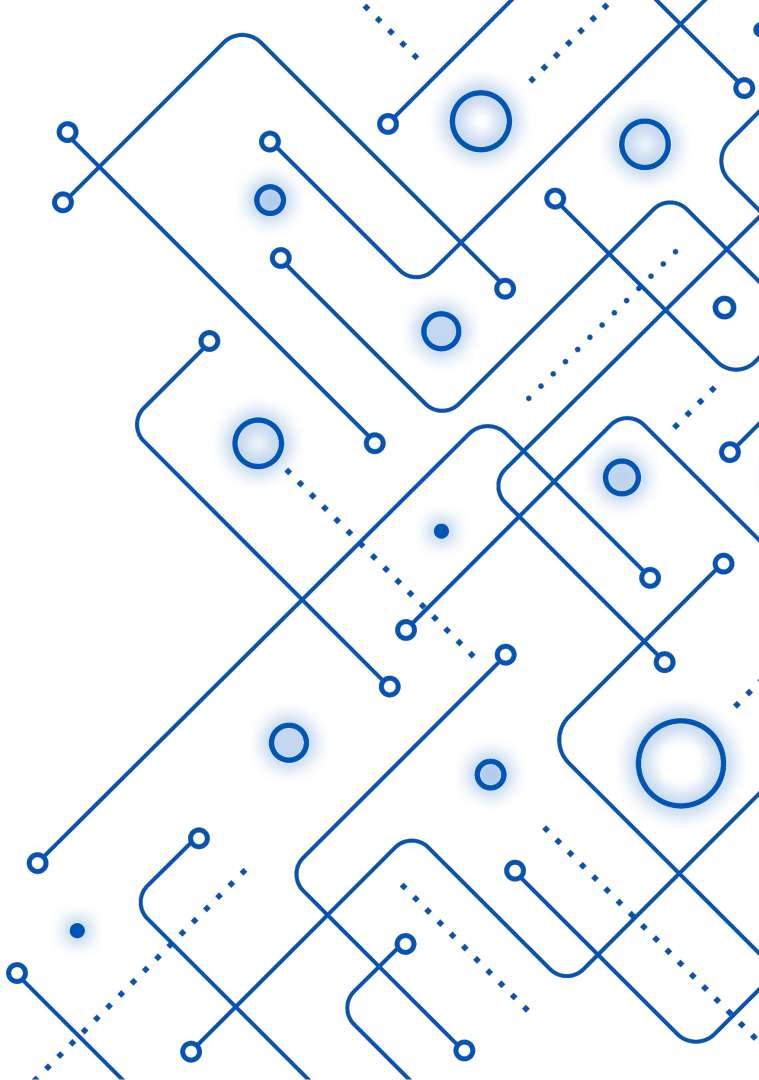

TerminusDB
The DataOps Revolution

DevOps + CI/CD has revolutionised software delivery



TerminusDB

CI/CD

Continuous Integration / Continuous Deployment

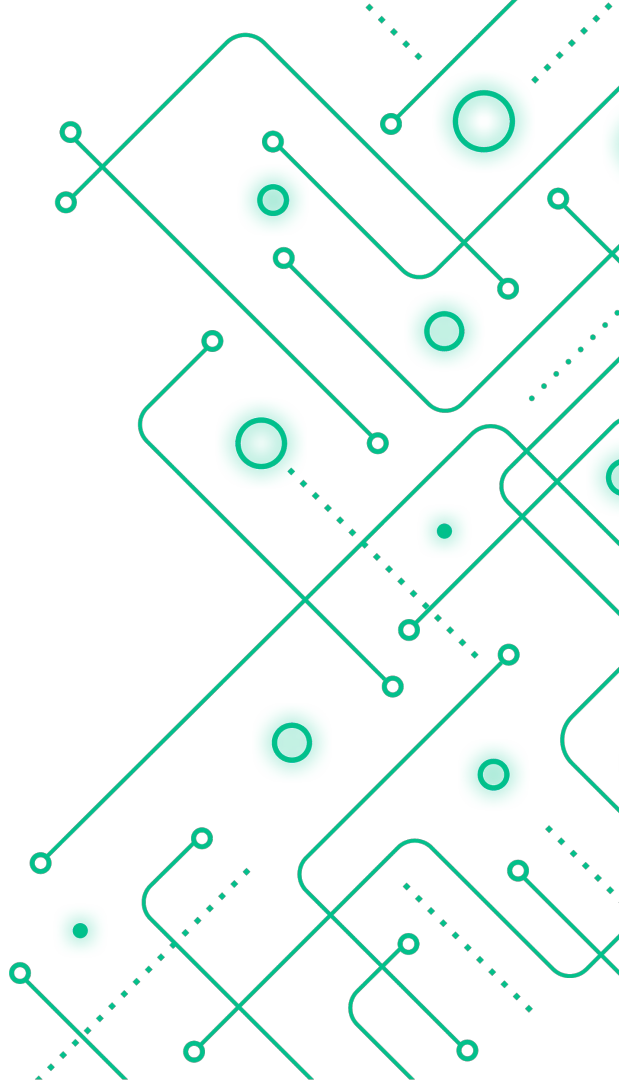
Pipelines with repositories

TerminusDB

Testing built into the pipeline

- We write code
- Check it in
- Run tests
- Merge tested code into dev
- Perform acceptance test
- Merge to master

And presto, deployed (via build, docker, packages whatever automagically)



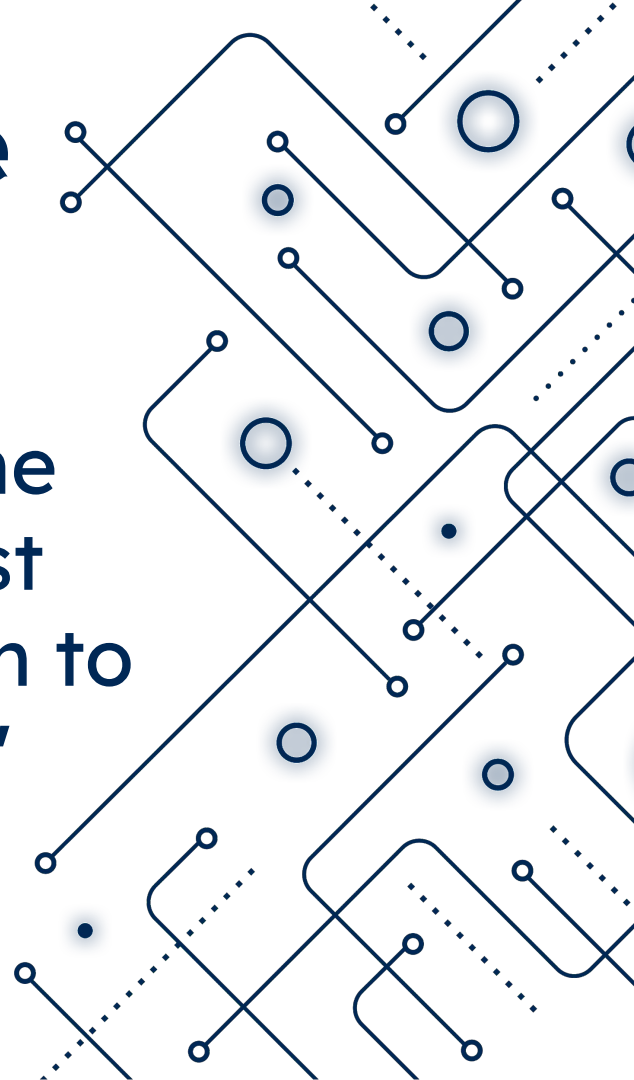
TerminusDB

But DataOps is still stone age

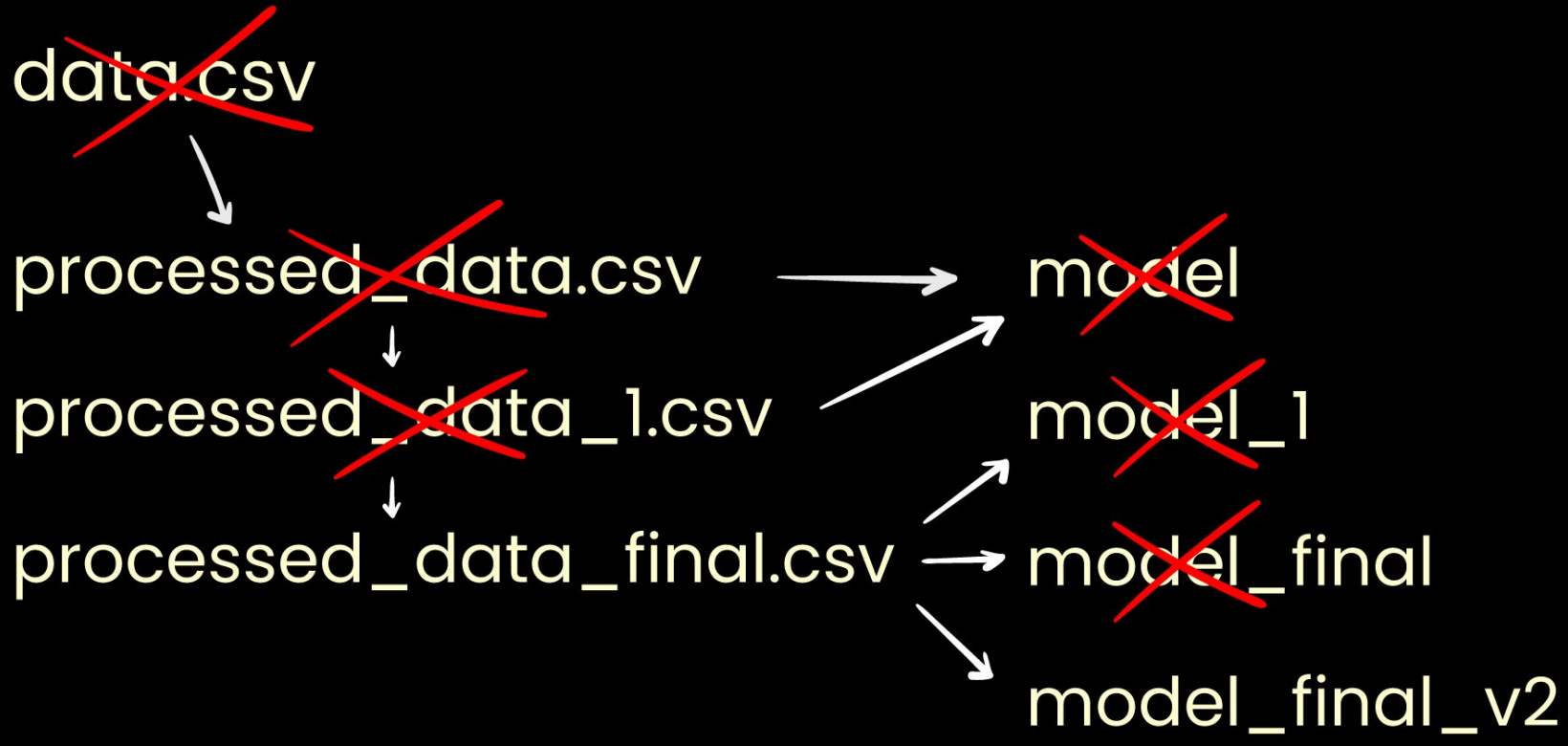
"Can you E-mail me the csv with the latest numbers?"

But DataOps is still stone age

"What were the results with the hyper parameters we used last week in the GAN? I can't seem to find the output on my laptop."



TerminusDB

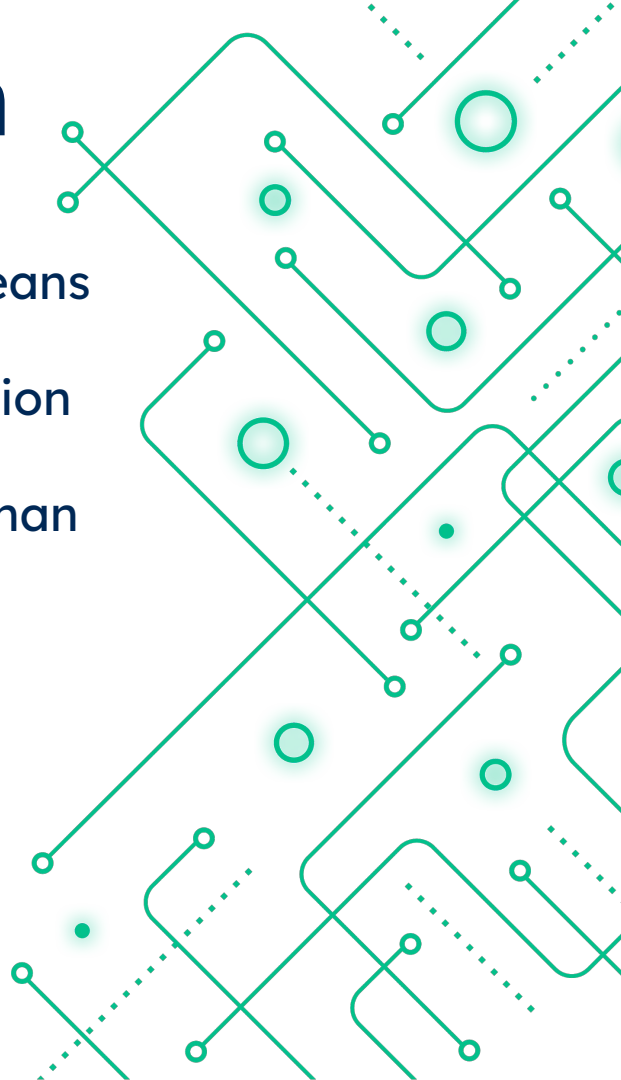

data.csv
processed_data.csv
processed_data_1.csv
processed_data_final.csv
model
model_1
model_final
model_final_v2

Knowledge Collaboration

- Transactional databases help provide a shared view of the world
- But they do not store revision information (automatically)
- They can't branch or clone easily
- They can't merge
- This makes pipelining data a real pain
- Shared data can mean shared destruction

Knowledge Collaboration

- We also want to understand what the data means when we are collaborating.
- A rich schema language provides documentation of how a data model works
- Schema extension in graphs is much simpler than in RDBMs's
    - New classes and properties can be added easily
- Also queryable/updateable in TerminusDB!



TerminusDB

Git is great for code, but not data

- A "line of code" makes some sense. Line of data as the granularity does not.
- Git is not efficient for storage or transmission of large datasets.
- Diffs are painful and not very meaningful.
- Git is not designed to be searchable - this is absolutely necessary for large datasets

ML requires DataOps

- Often many people working on different approaches to the same dataset
- Huge numbers of important tag enrichment on complex datasets.
- Often want to couple results with hyper-parameters
- Currently it is often done completely haphazardly with csvs.
- If you are lucky you get a database, but how to structure it? Do I just keep adding columns?
- And what the heck did we do last week?

TerminusDB

DataOps is not just for ML

It's for anyone who has structured data that evolves over time

TerminusDB is a Database

- Query the database to obtain the lastest information
- Update the shared view of the world
- Query back in time on old commits of the same database
- Branch an old commit and try something different
- Query the metadata about commits and branches to enable complex pipelining operations

TerminusDB

TerminusDB Is A Git for Data

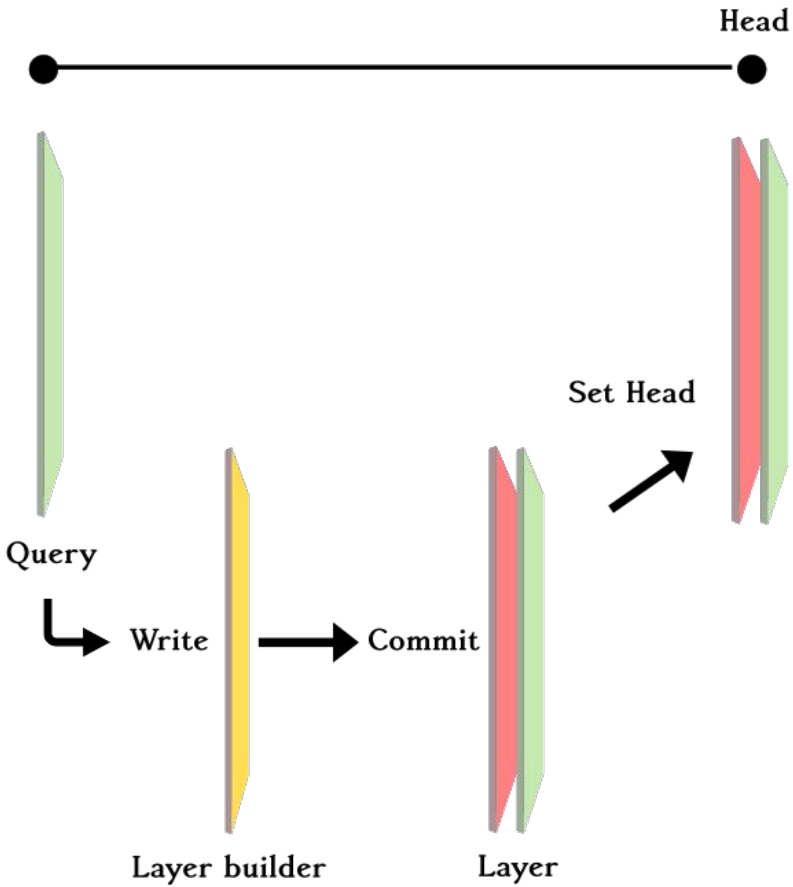
CI/CD can hit the database now.

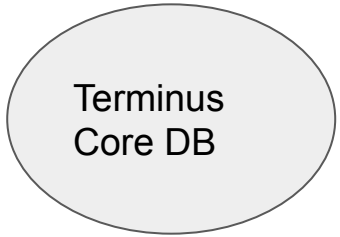
We allow you to clone, branch, merge, time-travel and roll-back in your data

TerminusDB

TerminusDB is a Timetravelling Database

- Query the database to obtain the lastest information
- Update the shared view of the world (head)
- Query back in time on old commits of the same database
- Branch the DB or an old commit of the DB and try something different
- Query the metadata about commits and branches to enable complex pipelining operations


Head
Set Head
Query
Write
Commit
Layer builder
Layer


Terminus
Core DB

TerminusDB Is A Git for Data

CI/CD can hit the database now.

In our TerminusHub release we will allow you to clone, branch, merge, time-travel and roll-back your data

TerminusDB

Give TerminusDB a try!

Terminusdb.com

https://github.com/terminusdb/terminus-quickstart