# AI Systems Support with Databases on the LRZ Compute Cloud

2023-12-01 | LRZ AI Training Series

# What is a Database and how can it be used with AI

- AI needs a lot of data (we know…)
- But what is data?
  - Images on your file system? *e.g.,* file:///home/john/datasets/MNIST/*
  - A  zip file shared with you? *e.g.,* https://www.dropbox.com/s/lrz%20Dataset.zip?dl=0
  - A public dataset? *e.g.,* ftp://io.erda.dk/dataset.h5
  - A built-in dataset? *e.g.,* from tensorflow.keras.datasets import fashion_mnist

- DB can be used to store, manage, and retrieve data and can be specifically designed for a certain type of data: Tabular, Time series, Graphs
  - Bonus: Data pre-processing / feature engineering (80-90% of “AI” work)
  - Bonus: Version control, backup, and archiving
  - Bonus: concurrency: Many clients at the same time

# What is a Database and how can it be used with AI

## Clients
## (Your Python code)

mysql://steve@10.100.100.10

SELECT age, gender FROM patients;

mysql://mark@10.100.100.10

SELECT * FROM train_images;

mysql://bill@10.100.100.10

DROP DATABASE do_not_delete;

## The DB server
## (C, C++, Rust)

A physical server
Or a VM on cc.lrz.de

10.100.100.10

## The DB file system

```
.
├── custom.cnf
├── databases
|   ├── aria_log.00000001
|   ├── aria_log_control
|   ├── ddl_recovery-backup.log
|   ├── ddl_recovery.log
|   ├── database
|   ├── ib_buffer_pool
|   ├── ibdata1
|   ├── ib_logfile0
|   ├── ibtmp1
|   ├── maria.err
|   ├── multi-master.info
|   ├── mysql
|   ├── mysql_upgrade_info
|   ├── performance_schema
|   └── sys
└── log
    └── mysql
```

# Use case: Graph Database

AI Support with Databases on the LRZ Compute Cloud

# Usecase: Clinical Knowledge Graph

What is the best way to organize your data?

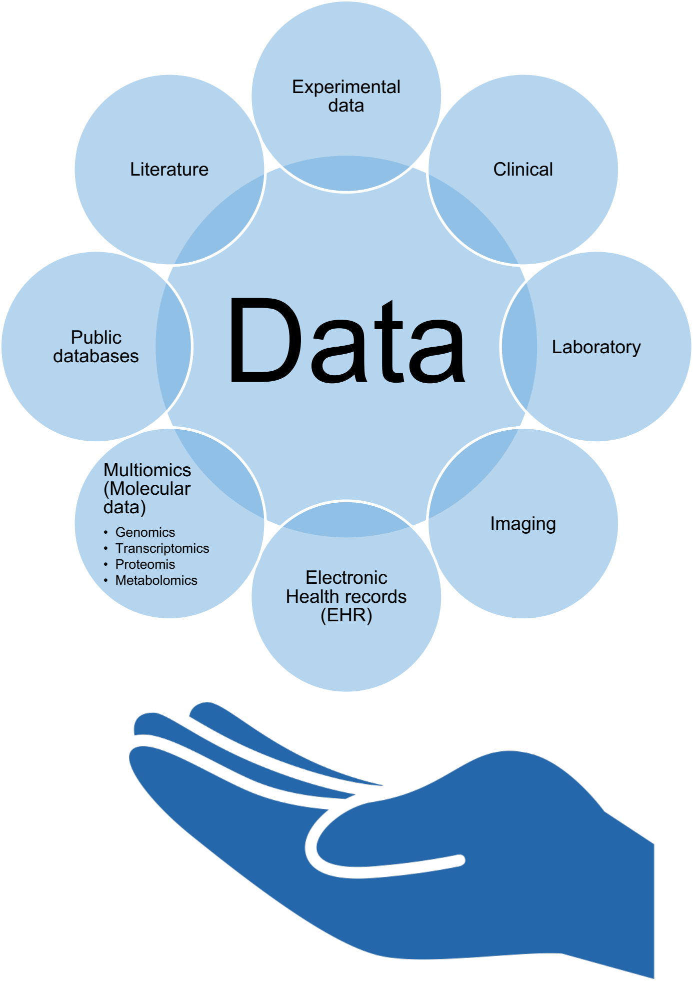
Idea: Use a DB that fits your data structure (relational, time series, vector etc.)

If it’s a graph of relationships: use a graph database then!

A. Santos *et al.*, “Clinical Knowledge Graph Integrates Proteomics Data into Clinical Decision-Making,” Bioinformatics, preprint, May 2020. doi: 10.1101/2020.05.09.084897.

## AI Support with Databases on the LRZ Compute Cloud

# Usecase: Clinical Knowledge Graph



Data

Experimental data

Clinical

Laboratory

Imaging

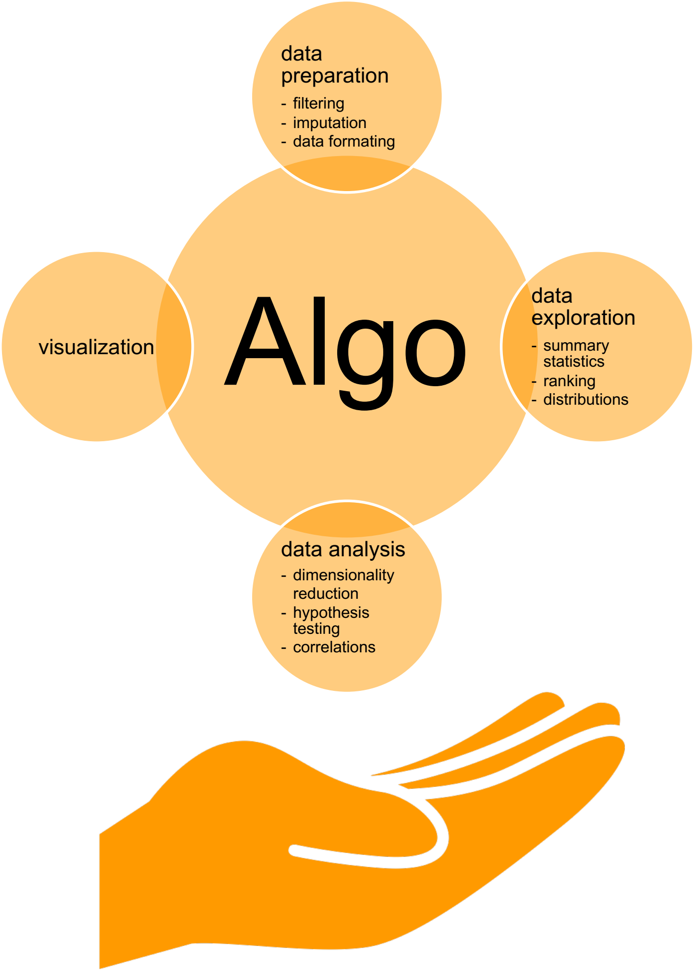
Electronic Health records (EHR)

Multiomics (Molecular data)
- Genomics
- Transcriptomics
- Proteomis
- Metabolomics

Public databases

Literature

Comprehensive representation of relevant data



Algo

data preparation
- filtering
- imputation
- data formating

data exploration
- summary statistics
- ranking
- distributions

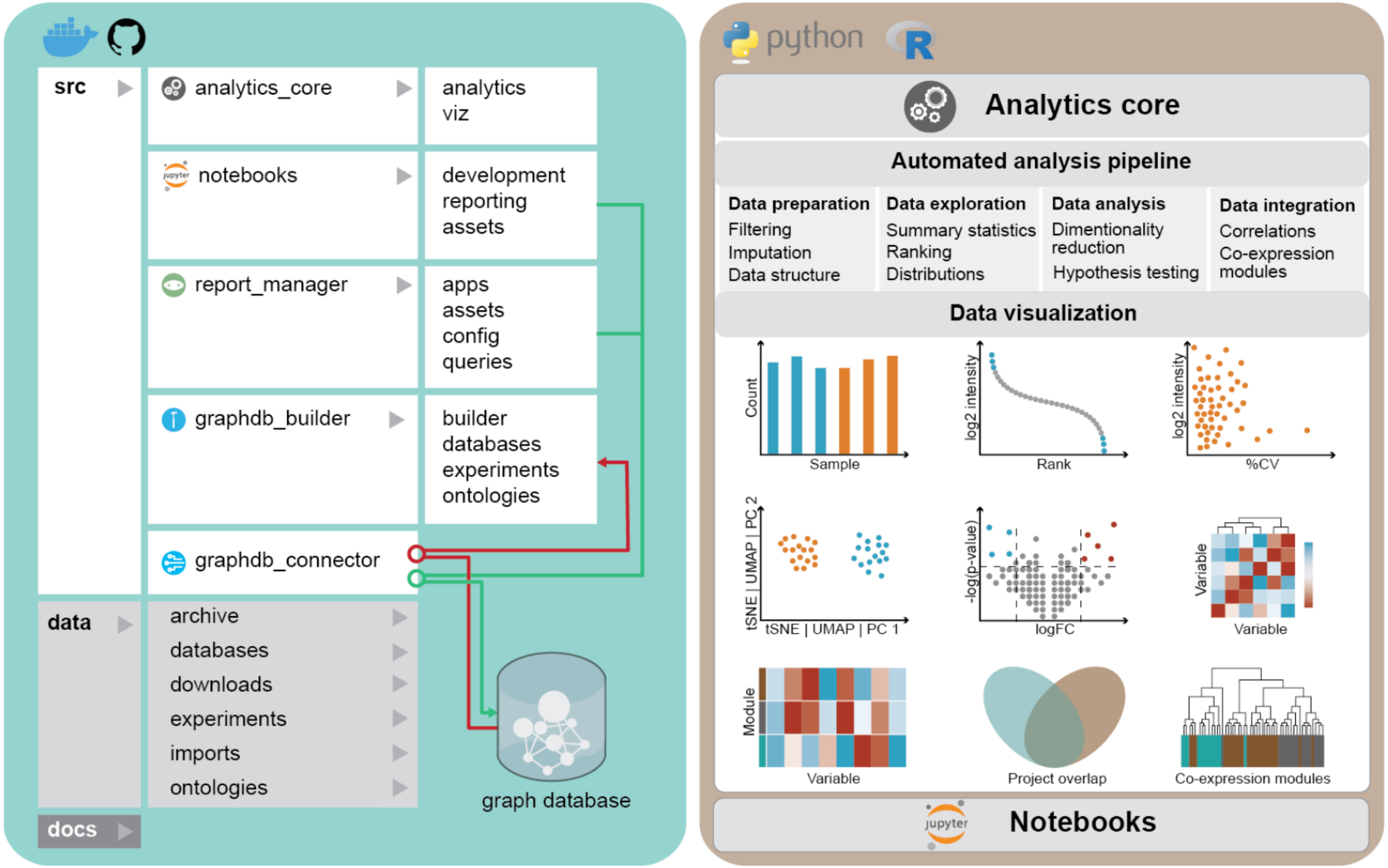
data analysis
- dimensionality reduction
- hypothesis testing
- correlations

visualization

Reproducible data processing

**Evidence based precision medicine**

# Usecase: Clinical Knowledge Graph



src

- analytics_core: analytics, viz
- notebooks: development, reporting, assets
- report_manager: apps, assets, config, queries
- graphdb_builder: builder, databases, experiments, ontologies
- graphdb_connector

data

- archive
- databases
- downloads
- experiments
- imports
- ontologies

docs

graph database

python R

Analytics core

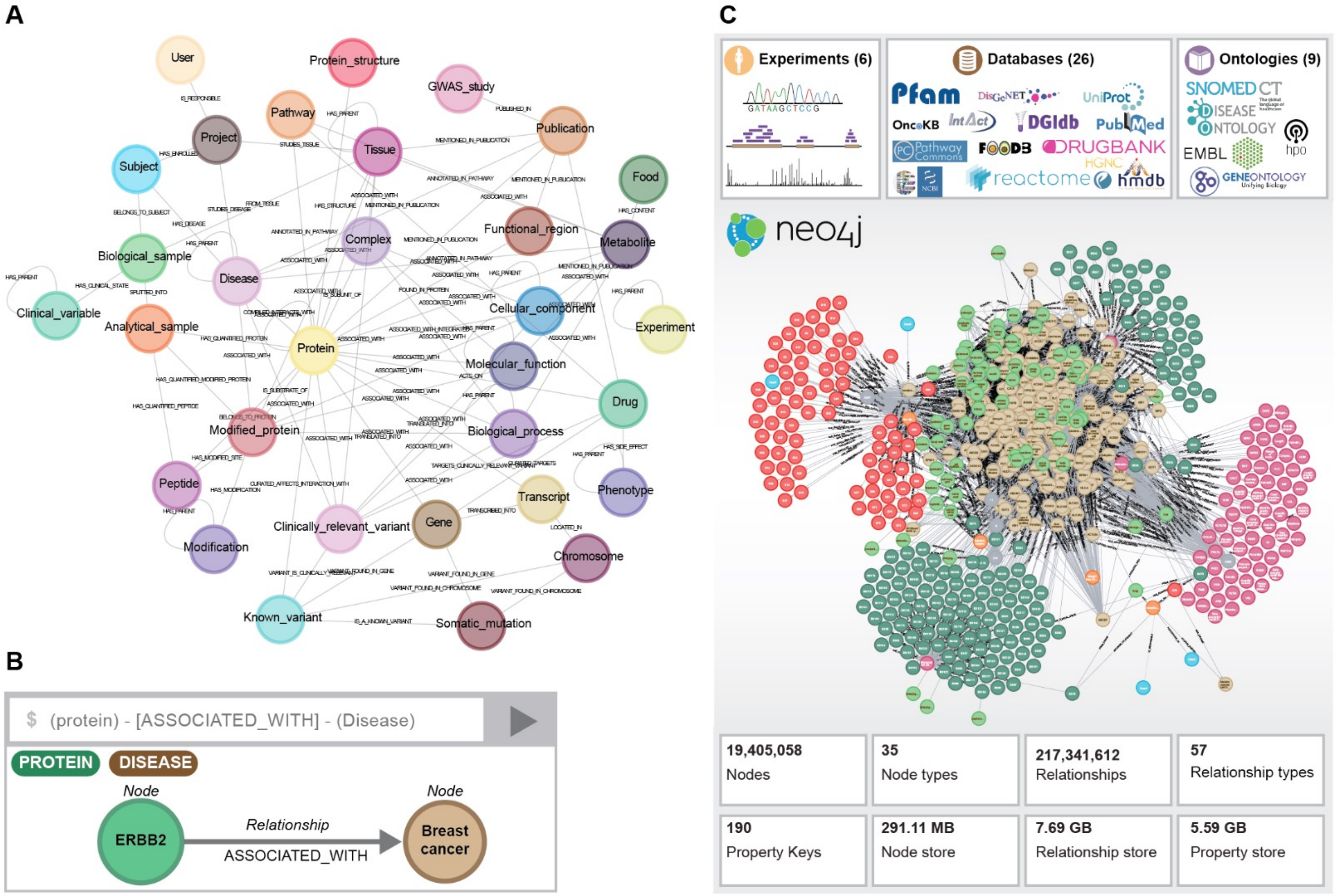
Automated analysis pipeline

| Data preparation | Data exploration | Data analysis | Data integration |
|---|---|---|---|
| Filtering<br>Imputation<br>Data structure | Summary statistics<br>Ranking<br>Distributions | Dimentionality reduction<br>Hypothesis testing | Correlations<br>Co-expression modules |

Data visualization

Notebooks

# Usecase: Clinical Knowledge Graph

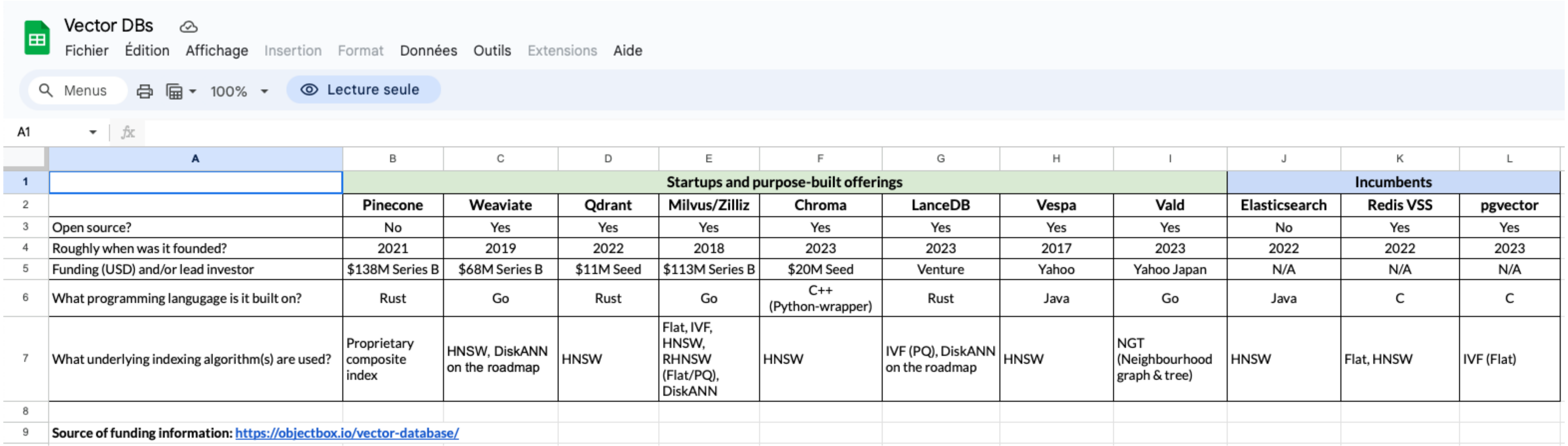


```
$ (protein) - [ASSOCIATED_WITH] - (Disease)
```

| | | | |
|---|---|---|---|
| 19,405,058<br>Nodes | 35<br>Node types | 217,341,612<br>Relationships | 57<br>Relationship types |
| 190<br>Property Keys | 291.11 MB<br>Node store | 7.69 GB<br>Relationship store | 5.59 GB<br>Property store |

# Use case: Vector Database

# Usecase: Vector Databases



| | Startups and purpose-built offerings | | | | | | | | Incumbents | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Pinecone** | **Weaviate** | **Qdrant** | **Milvus/Zilliz** | **Chroma** | **LanceDB** | **Vespa** | **Vald** | **Elasticsearch** | **Redis VSS** | **pgvector** |
| Open source? | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Roughly when was it founded? | 2021 | 2019 | 2022 | 2018 | 2023 | 2023 | 2017 | 2023 | 2022 | 2022 | 2023 |
| Funding (USD) and/or lead investor | $138M Series B | $68M Series B | $11M Seed | $113M Series B | $20M Seed | Venture | Yahoo | Yahoo Japan | N/A | N/A | N/A |
| What programming langugage is it built on? | Rust | Go | Rust | Go | C++ (Python-wrapper) | Rust | Java | Go | Java | C | C |
| What underlying indexing algorithm(s) are used? | Proprietary composite index | HNSW, DiskANN on the roadmap | HNSW | Flat, IVF, HNSW, RHNSW (Flat/PQ), DiskANN | HNSW | IVF (PQ), DiskANN on the roadmap | HNSW | NGT (Neighbourhood graph & tree) | HNSW | Flat, HNSW | IVF (Flat) |

**Source of funding information:** https://objectbox.io/vector-database/

# Usecase: Vector Databases

- **Definition**: Databases optimized for storing and querying high-dimensional vectors.
- **Use Cases**:
  - Similarity search in embedding spaces.
  - AI model feature storage and retrieval.
  - Content-based recommendation systems.
- **Efficiency**: Enables near real-time search for the nearest vectors in large-scale datasets.
- **Indexing Mechanisms**: Utilizes specialized data structures to allow efficient similarity searches.
- **Integration with ML Frameworks**: Compatible with vector embeddings from popular AI frameworks like TensorFlow, PyTorch, and more.
- **Scaling**: Supports distributed architectures for handling billion-scale vector datasets

# Usecase: Vector Databases

| id | first_name | last_name | country |
|---|---|---|---|
| 1 | François | Dubois | FR |
| 2 | Juanita | García | ES |
| 3 | Ursula | Ottovordemgentschenfelde | DE |
| 4 | Jun | Wang | CN |

Relational database

Vector database

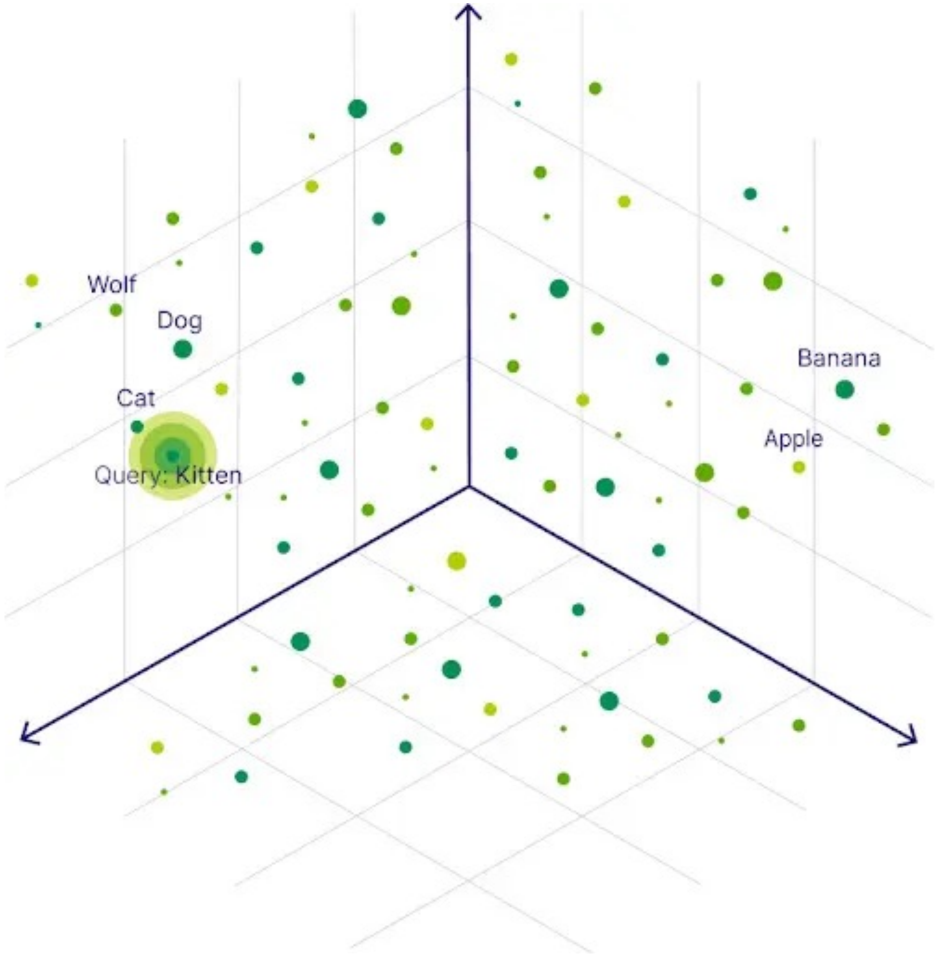
| text | embedding |
|---|---|
| “Ceci n’est pas une pipe” | [0.665, 0.874, 0.002 … 0.873, 0.112] |
| “Aimer c’est décevoir un peu” | [0.865, 0.004, 0.542 … 0.887, 0.136] |
| “Paris c’est magique pour ceux qu’ont du biff de té-cô” | [0.963, 0.774, 0.102 … 0.830, 0.812] |
| “Les silences comptent aussi” | 666, 0.174, 0.082 … 0.425, 0.999] |

ML

*Vector databases store and provide access to structured and unstructured data, such as text or images, alongside their vector embeddings.*

# Usecase: Vector Databases

- Vector embeddings are the data’s numerical representation of its semantic meaning.
- Idea: use ML model to generate the vector embeddings (feature extraction)
- Similar objects are close together in vector space → can be calculated based on the distance between the data object’s vector embeddings



https://weaviate.io/blog/what-is-a-vector-database

# Demo time

# Usecase: Vector Databases

## How to perform image similarity search?

1. Acquire images
2. Acquire a ML model(ResNet18 for us)
3. Feed it our images = for each we get a vector representing the image
4. Index images along with their vectors in a database (milvus for us)
5. Leverage database for similarity search

*What images are semantically similar to this bad boi in my dataset?*

https://github.com/dshvimer/milvus-up-and-running

# Usecase: Vector Databases

1. Create security group, VM, floating IP
2. git clone https://github.com/flrntdfr/milvus-demo.git
3. cd milvus-demo
4. sudo bash run.sh
5. open URL

https://github.com/dshvimer/milvus-up-and-running