Kùzu
A Graph Database Management System for Python Graph Data Science

https://kuzudb.com
Outline

● What are graphs?
● When do you need graph modeling?
● Features of a competent graph database management system
● Kùzu’s vision
  ○ As a GDBMS
  ○ As the go-to solution for graph data science
● Walkthrough: How Kùzu makes graph data science easier
What are graphs/networks?

Abstract representation of entities and relationships
Graphs: Natural ways to represent data

Social network

Drug interactions

Traffic networks

Transaction graph

Molecular networks

Knowledge graphs
Graph DBMS: Overview

Data Models
Labelled Property Graph

```
MATCH (a:account)-[Transfer*]->(b:account)
WHERE a.name=Bob
RETURN b.name
```

Query Language
High level query language designed for graphs

Systems
Graph-specific storage structures, indices, operators

RDF
Subject - Predicate - Object

Triple
When do you need graph modeling?

Recursive joins, path-finding and identifying patterns

“Give me all direct or indirect possible sources of money flow into Alice’s account from Canada.”

MATCH (a:Account)-[:Transfer*]->(b:Account)
WHERE a.location="Canada" AND b.owner="Alice"
RETURN *

Recursive SQL? Hard!

“Give me shortest path of money flow into Alice’s account from Canada.”

MATCH (a:Account)-[:Transfer* SHORTEST]->(b:Account)
WHERE a.location="Canada" AND b.owner="Alice"
RETURN *

Recursive SQL? Super Hard!
When do you need graph modeling?

Very heterogeneous datasets to construct “knowledge graphs”
Feature set of a competent graph database

What every competent GDBMS should do:

- Pre-defined/pointer-based joins: joins are defined between nodes through edges
- Many-to-many growing joins
- Heterogeneous datasets (e.g., knowledge graphs)
- Recursive join queries
  ```
  MATCH (a:Account)-[:Transfer*]->(b:Account)
  ```
- Schema querying
  ```
  MATCH (a:Account)-[e1]->(b:Account)-[e2]->(c:Account)
  WHERE type(e1) != type(e2)
  RETURN *
  ```

Kuzu Vision: GDBMS

Kuzu aims to represent the state-of-the-art of how graphs should be stored, indexed and queried

- **Highly scalable** to several TBs of data
- **Very fast** query speeds
- **Property graphs** + RDF support via Cypher query language
- **Easy to use** & embeddable (like DuckDB/SQLite, but for graphs)

Kuzu is based on many years of research at University of Waterloo. It’s now being developed at a spin-off company called Kuzu Inc.
Kùzu Vision: Graph Data Science

To be the go-to backend for graph modeling and data science

Graph Learning

Data science ecosystem

Graph Analytics & Viz

Interop formats

Data lakes

RDBMS
How Kùzu makes Graph Data Science easier

- 2 data sources w/ people, their friends & movies they watch
- Goal: Movie recommender system
- Approach: Graph Neural Network (GNN)-based link prediction

```
<table>
<thead>
<tr>
<th>Person</th>
<th>Name</th>
<th>Age</th>
<th>Watched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>25</td>
<td>m1</td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>20</td>
<td>m2</td>
<td></td>
</tr>
<tr>
<td>Carl</td>
<td>30</td>
<td>m3</td>
<td></td>
</tr>
</tbody>
</table>

Dataset 1

```

```
<table>
<thead>
<tr>
<th>Person</th>
<th>Friend</th>
<th>LivesIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Bob</td>
<td>Toronto</td>
</tr>
<tr>
<td>Bob</td>
<td>Carl</td>
<td>Waterloo</td>
</tr>
<tr>
<td>Carl</td>
<td></td>
<td>Waterloo</td>
</tr>
</tbody>
</table>
```

Dataset 2
A very brief overview of graph ML

- Graph ML goal: Embed nodes into a vector space
- Benefits: Models incorporate the structure of a graph based on its relationships
How Kùzu makes Graph Data Science easier

Data might exist in multiple sources

Person
Movies
Watched

Person
Friends
LivesIn

Person

Movies

Watched

Person
Friends
LivesIn
How Kùzu makes Graph Data Science easier

Deduplication + remove dangling nodes

```
M1
n: Alice  a: 25
n: Bob    a: 20
n: Carl   a: 30
```

```
M2
Watched
```

```
M3
Watched
```

```
W
Friends
LivesIn
```

```
T
Friends
LivesIn
```

```
M1
Watched
```

```
M2
Watched
```

```
M3
Watched
```

```
W
```

```
T
```

```
Dan
```

```
LivesIn
```

```
Friends
```

```
LivesIn
```

```
LivesIn
```

```
W
```

```
T
```

```
W
```

```
T
```

```
W
```

```
M3
```

```
Dan
```

```
Friends
```

```
Friends
```

```
M1
```

```
M2
```

```
M3
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```

```
W
```
Steps in graph learning pipeline with Kùzu

1. Install
2. Load data to a graph
3. Deduplication
4. Remove dangling nodes
5. Export to PyG
6. Train GNN
7. Make predictions

How Kùzu makes Graph Data Science easier

```python
pip install kuzu
import kuzu
...
conn.execute('CREATE NODE TABLE person ...
conn.execute('COPY person FROM ...
...```


Steps in graph learning pipeline with Kùzu

1. Install
2. Load data to a graph
3. Deduplication
4. Remove dangling nodes
5. Export to PyG
6. Train GNN
7. Make predictions

Once data and features exist are loaded to a graph, work in PyG as normal
How Kùzu makes Graph Data Science easier

Graph learning pipeline

1. Kùzu: Data ingestion from PostgreSQL, CSV, and Parquet formats.
2. Data processing and analysis using Kùzu.
3. Graph model building: GNN (Graph Neural Network).
4. Predictions: Recommendations or other predictions based on the GNN model.
5. Output: Predictions and GNN model.
Takeaways

● Kùzu is an in-process, analytical graph database system
  ○ “Like DuckDB/SQLite, but for graphs”
● Highly scalable: optimized for multi-core parallelism
● Integrated with the PyData ecosystem: numpy, pyarrow, NetworkX, PyTorch
● Support for property graphs and RDF graphs via Cypher query language
● Embeddable and easy to use from within your application
Kùzu is **open source** (MIT licensed)
Give it a try and star us on GitHub!

github.com/kuzudb/kuzu

Join our Discord!

@kuzudb

https://kuzudb.com/blog/