

Financial Crime Detection using Graph & AI

Driving Innovation with Connected Data

Agenda

- Evolution of Financial Crime
- Relational DB in Fraud Detection
- The Solution – Graph
- Graph Algorithms & Data Science
- Graph in Fraud Detection
- Usecases

Evolution of Financial Crime

Financial crime is evolving—fraudsters are no longer acting alone but operating in complex, interconnected networks. Traditional rule-based systems often fail to uncover these hidden relationships, allowing fraudulent activities to go undetected.

Global Stats:

- Fraud losses (US) was over \$12.5 billion to fraud in 2024, up 25% from the previous year.
- Cyber fraud (India): High-value cases surged 4x in 2024, causing \$20 million in losses.
- Crypto crime: Illicit transactions surpassed \$40 billion in 2024, with projections exceeding \$51 billion.
- Carding & identity fraud: Synthetic identity fraud is projected to cause \$43 billion in losses by 2026.
- Money laundering and financial fraud is a problem that has become even harder to track with the proliferation of real-time digital transfers and payments.

Relational DBs in Financial Fraud Detection

Financial data generally comprises of multiple tables. As data scales, joins become more expensive between the tables.

1. Poor performance with **complex joins**.
2. SQL databases follows **rigid schema and row-based storage** which struggles to find hidden connections.
3. **Lack of real-time analytics** and capability to analysis vast and dynamic datasets.
4. Scalability issues with increase of data
5. Difficulty in handling **multi-hop** complex queries
6. Visualization is limited to tabular data
7. Inability to perform behavioral analysis Detect Anomalies Efficiently
8. Financial data is stored in separate tables to make it hard to find hidden fraud rings.

Relational DB vs Graph DB

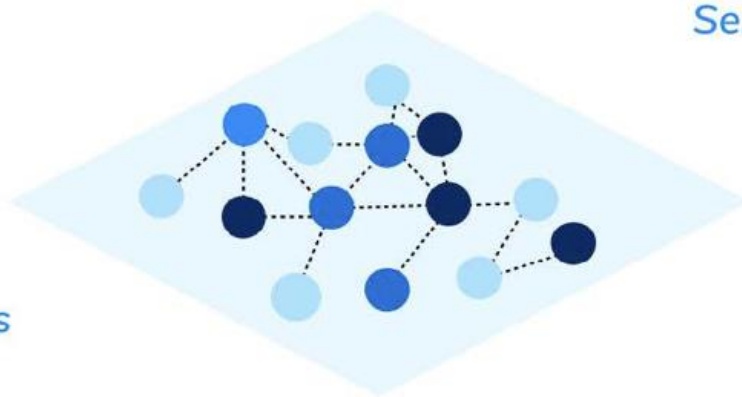
Aspect	Relational Databases	Neo4J
Data Model	Tables, rigid schema	Nodes & relationships, flexible schema
Deep Analysis	Slower for multi-level relationships (joins). Stores the data in silos.	Fast for complex queries, no expensive joins. Stores the data with relationships.
Query Speed	Multiple joins makes query more slow	Querying relationship is fast and simple
Schema Flexibility	Predefined, hard to modify	Dynamic, evolves with data
Generate Insights	Slow (batch processing)	Suitable for real time analysis
Visualization	Tabular data	Interactive Graph based
Query Language	SQL, complex for deep joins	Cypher, GQL, optimized for relationships



Data



Relationships

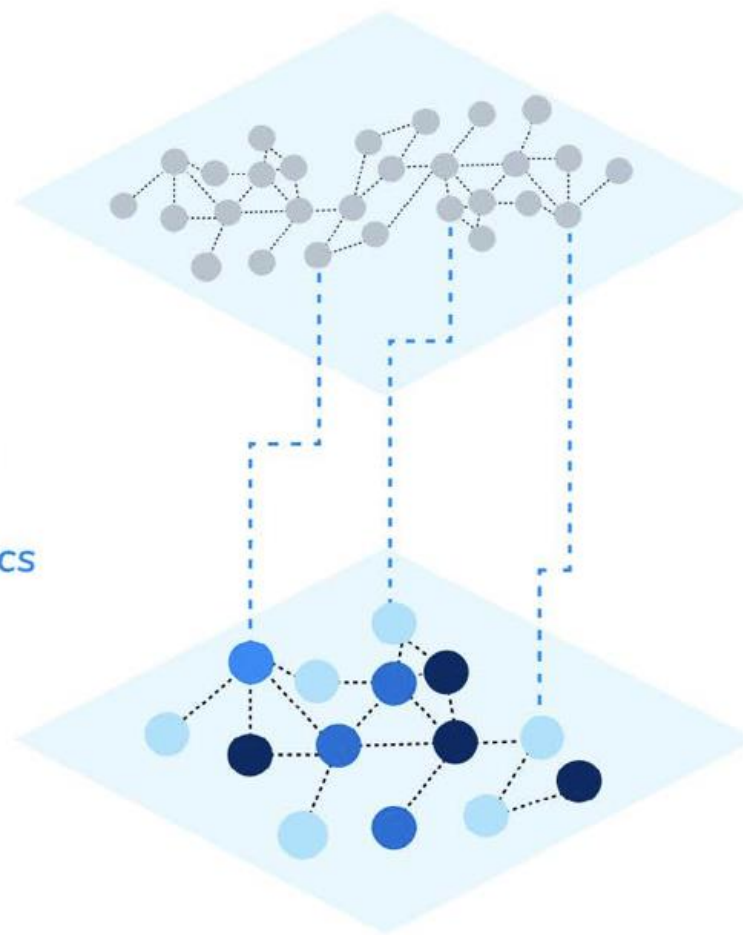


Graph

Dynamic Context



Semantics

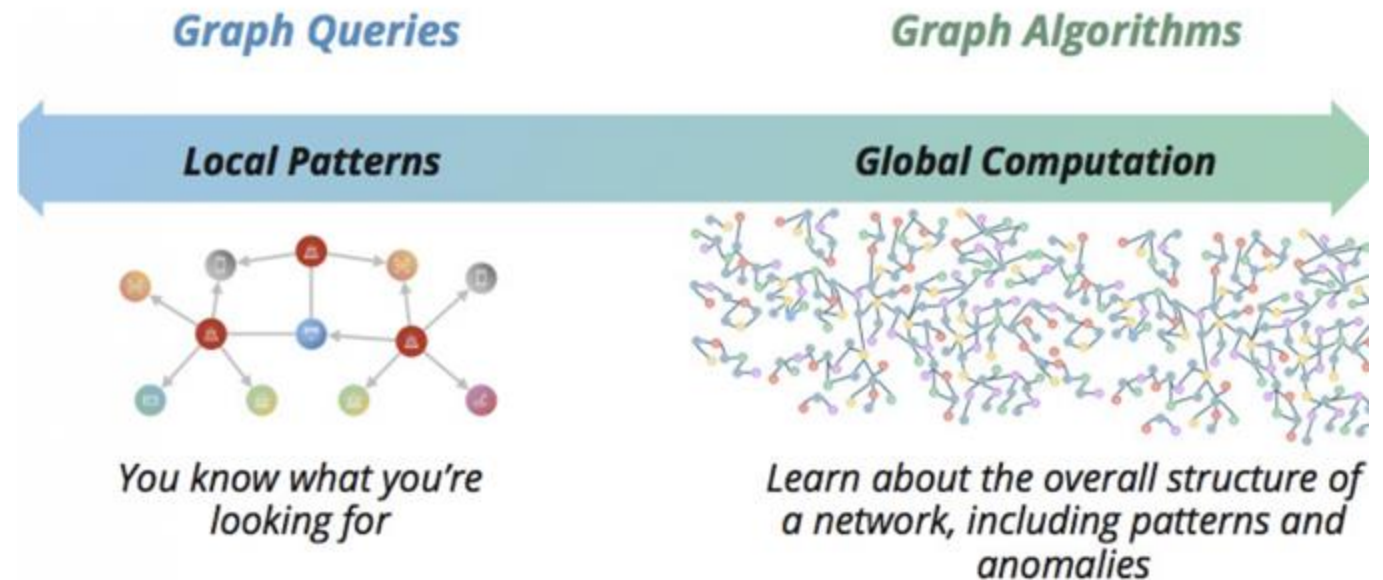


Knowledge Graph

Deep Dynamic Context

Solutions Graph provides

- ❖ **Fast and Scalable Fraud Detection**
 - Identify hidden fraud rings by mapping relationships in real-time.
- ❖ **Advanced Analytics & Machine Learning** – Use graph algorithms like **PageRank, Community Detection, and Similarity Measures** to flag suspicious activities.
- ❖ **AML Compliance & Risk Assessment** – Detect money laundering patterns, track fund flows, and ensure regulatory compliance.
- ❖ **Reactive fraud detection to a proactive risk mitigation strategy**



Graph Algorithms



Fraud Detection



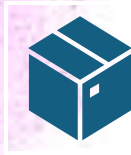
**Anti Money
Laundering
(AML)**



**Assessing
Investment Risk**



Cybersecurity



**Data Privacy &
KYC**



**Entity De-
Duplication**



Graph ML



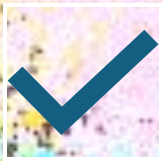
**Customer 360
View**



MDM



**Regulatory
Compliance**



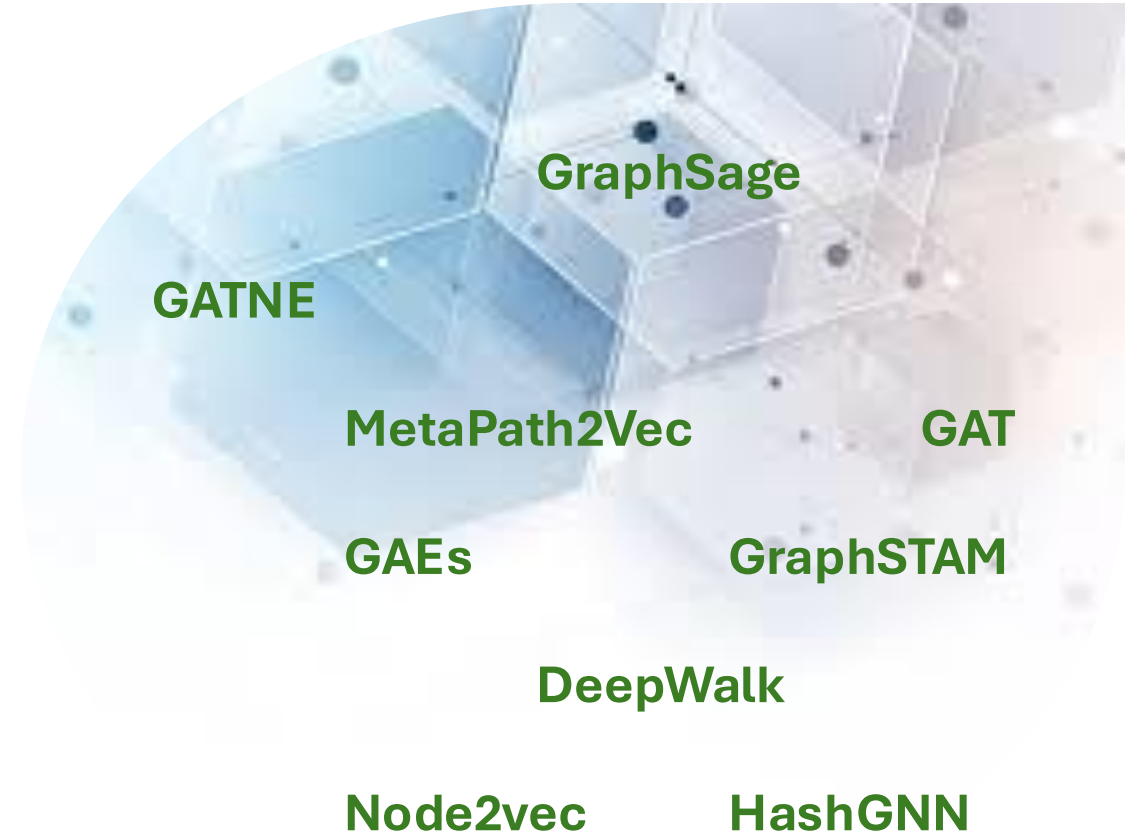
**Product
Recommendation**



**Predictive
Maintenance**



**Fraudulent Call
Detection**



Use Cases

Anti Money Laundering

Neo4J can catch money launderers by tracking suspicious patterns of transactions

- Banks are under increasing pressure to identify attempts by banned groups to move money surreptitiously and risk huge fines if they fail.
- Neo4J can find mule accounts, money cycles and collusive behaviour in huge banking datasets by applying specialised network analytics algorithms from its graph data science library.
- Significantly more suspicious behaviour is identified, and the same algorithms can make it much easier for investigations teams to do their job.
- The algorithms can be tailored to the situation, and it is fast enough to run compute-intensive AML algorithms.



Fraud Detection

Neo4J can identify networks of fraudsters hidden in your data

- Banks struggle to identify networks of criminals defrauding them because any individual criminal looks little different to a regular customer when seen in isolation.
- Neo4j increases the accuracy of fraud detection by augmenting their existing machine learning detection systems with features that highlight how criminals are connected.
- Identifying more criminal networks not only reduces financial losses, but improves customer satisfaction because fewer genuine transactions are mis-labelled as suspicious.
- Neo4J is fast enough to be able to run the computationally intensive algorithms required over large banking datasets.



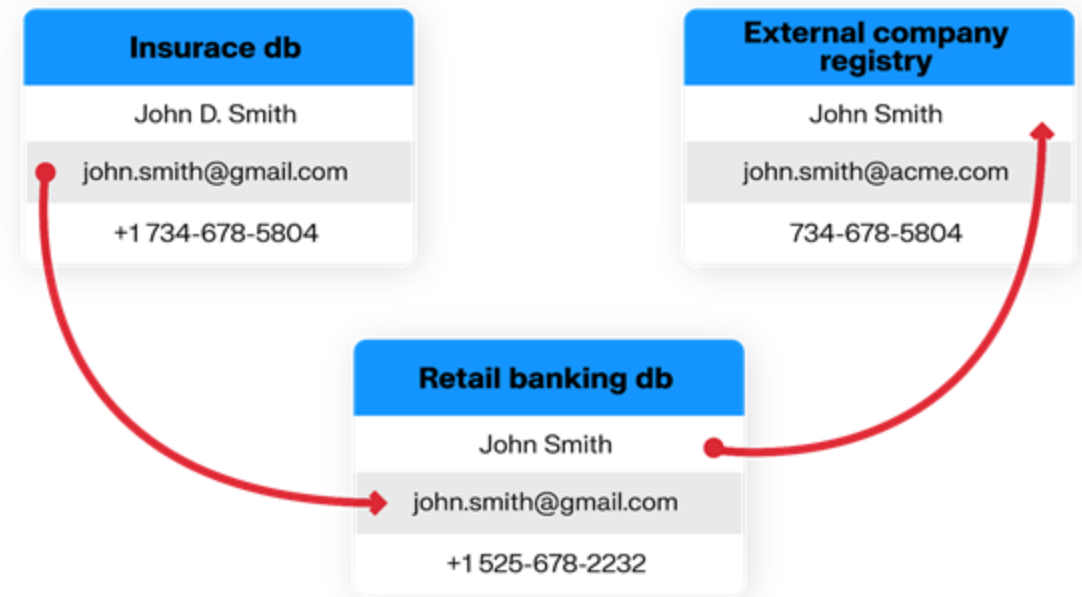
Entity De-duplication

Neo4J can identify and merge duplicate customer/account records.

Entity Deduplication eliminates redundancy caused by typos, multiple registrations, or fraud attempts. It uses graph-based relationships to detect hidden links between entities.

Applications -

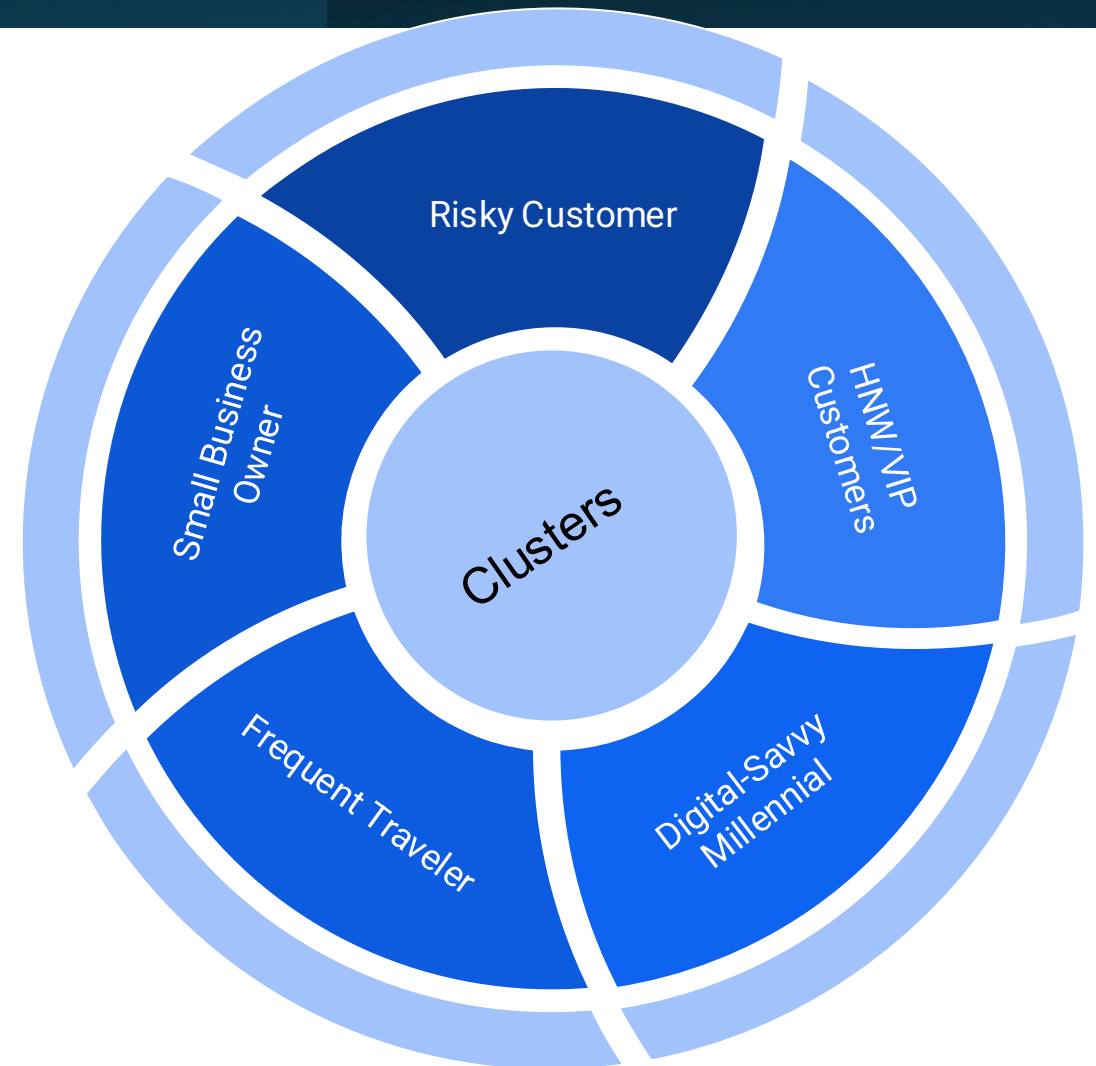
- **Fraud Detection:** Identify fraudsters using multiple identities.
- **Customer 360 View:** Consolidate customer data across multiple accounts.
- **Credit Risk Assessment:** Avoid duplicate loan applications under different names.
- **Regulatory Compliance:** Ensure KYC/AML rules are met with clean customer records.
- **Operational Efficiency:** Reduce manual efforts in reconciling duplicate profiles.



Customer Persona

Neo4J can identify networks of fraudsters hidden in your data

- Demographic Profile
- Graph-Based Red Flags
 - Multiple accounts linked to a single device/IP.
 - Connections to known fraudsters.
 - Frequent, circular transactions with no business rationale.
- High-value deposits & investments
- Prefers mobile banking & digital wallets
- Uses multiple business accounts for different venture
- Uses multi-currency accounts
- Product Usage
- Travel Card with airline miles.
- Value based
- Channel Preference



Helping solve the world's most challenging problems across every industry

Financial Services



Telco



Retail



Manufacturing



Life Sciences



Technology



Logistics



Government

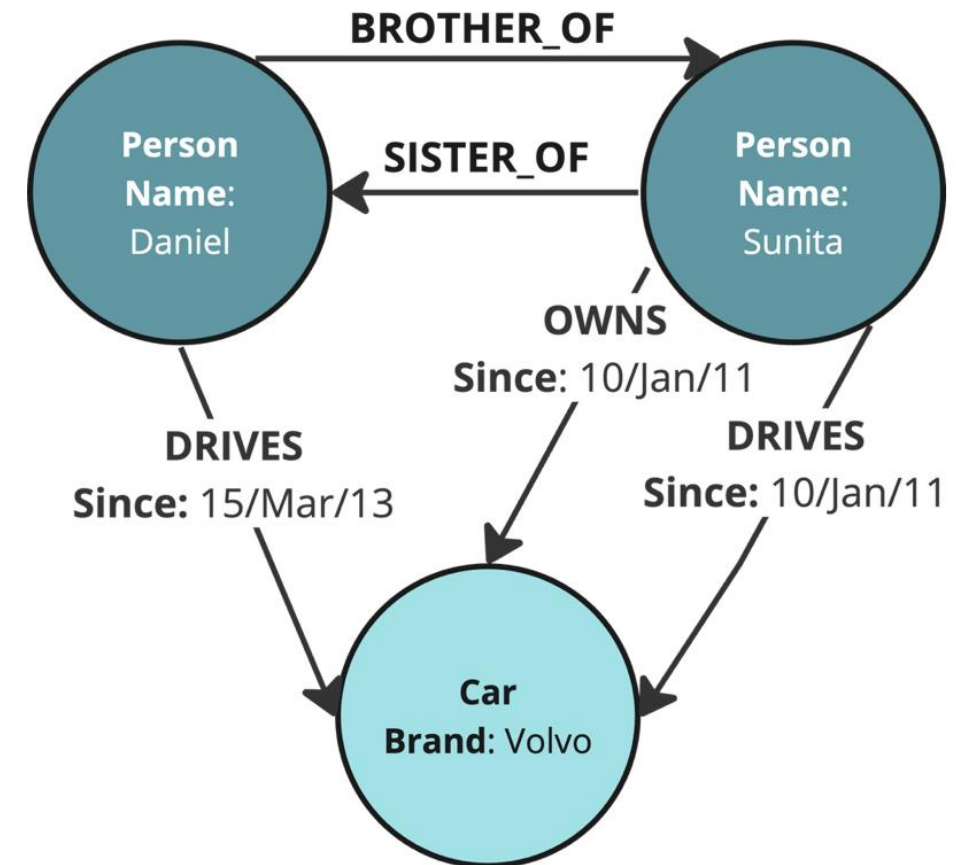
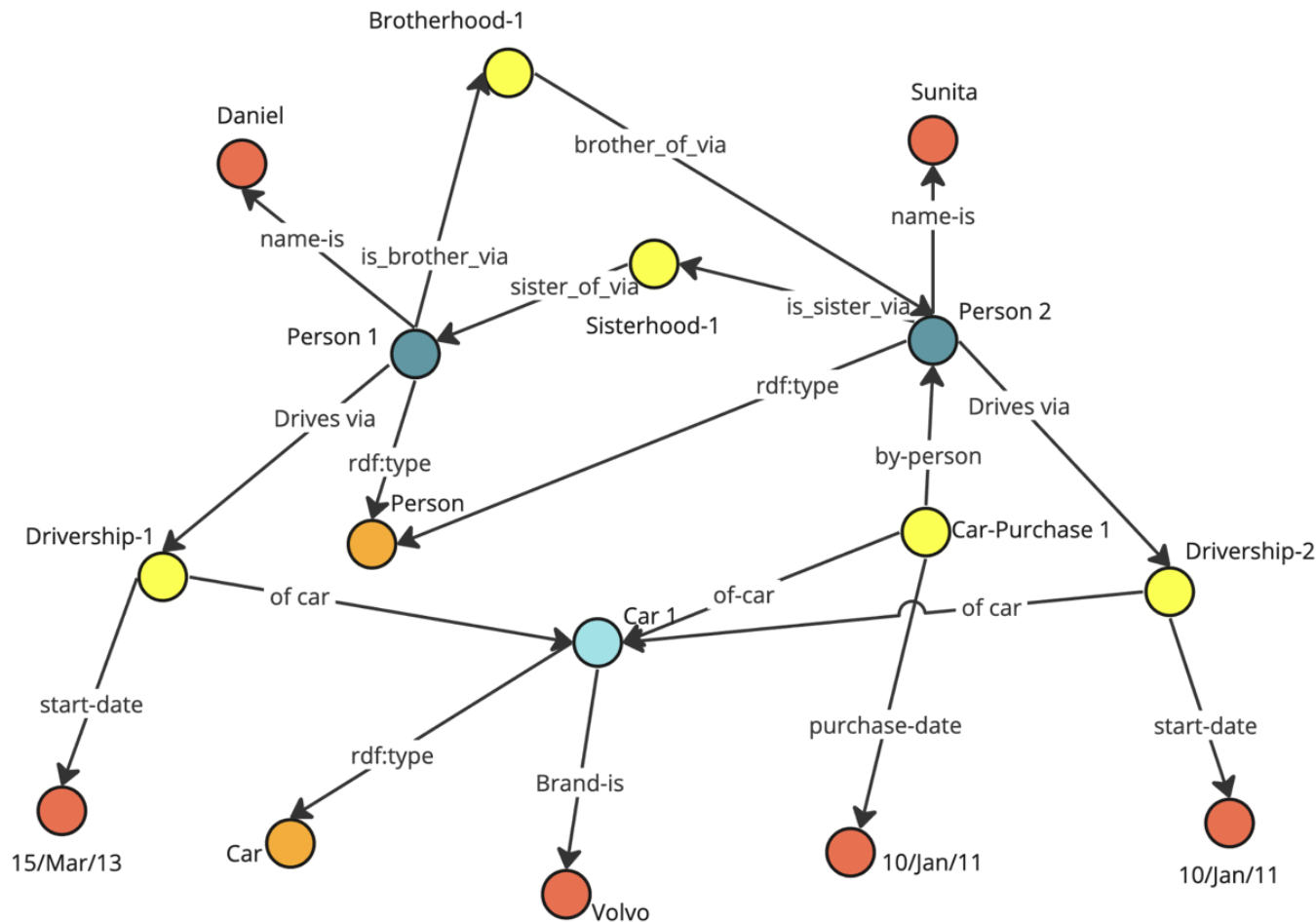


Thank you



Appendix






RDF vs Property Graph



Graph Algorithms

Neo4j Graph Data Science Library includes enterprise scalable graph algorithms optimized to run against connected data in Neo4j.

An enterprise-grade method for data scientists and analysts to run graph algorithms with connected data at scale.

Type of Graph Algorithm	Example Algorithms	Use in Fraud Detection
 Community Detection	Weakly Connected Components (Union Find), Louvain Modularity, Label Propagation	Identify disjointed groups that share identifiers. Identify communities that frequently interact.
 Similarity	Node Similarity using Jaccard	Measure account similarity or fraud ring similarity.
 Centrality	PageRank	Measure influence and transaction volumes.
 Heuristic Link Prediction	Common Neighbors	Find unobserved relationships and add them to your data.
 Pathfinding & Search	Shortest Path	Filter transactions with extremely short paths between people.