TigerGraph

T1
Graph Concepts

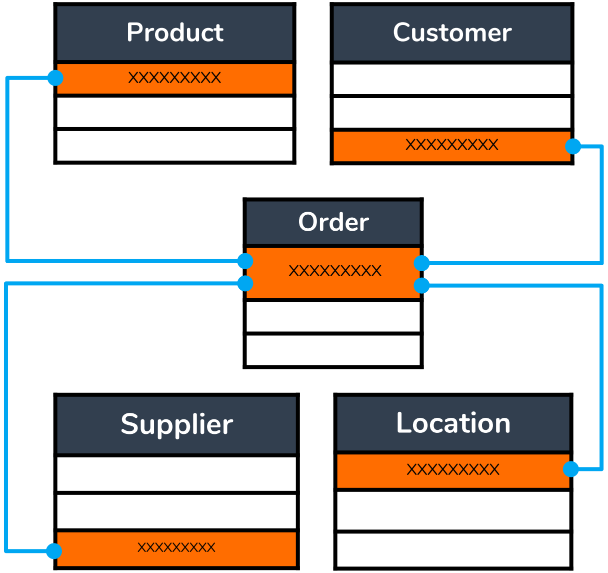
Szilard Barany
Principal Sales Engineer

Agenda

- Evolution of databases
- Evolution of graph databases
- Graph concepts
- Data science capabilities of a graph analytical platform

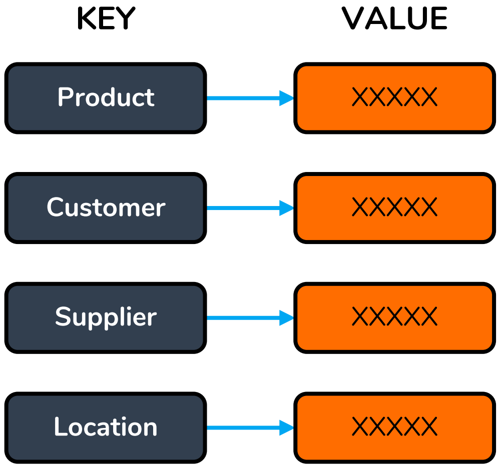
The Evolution of Databases

Relational Database



- Rigid schema
- High performance for transactions
- Poor performance for deep analytics

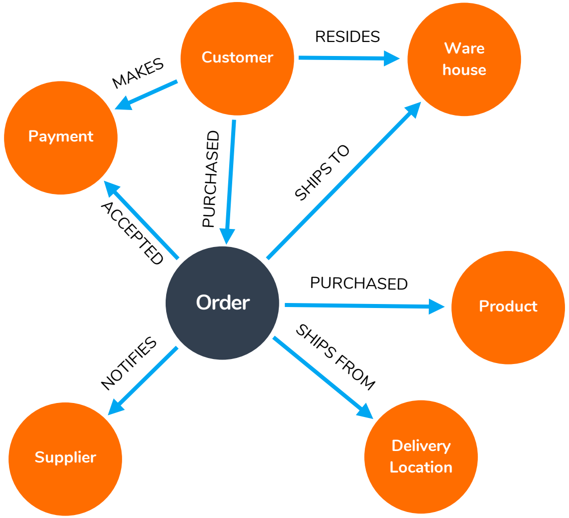
Key-Value Database



KEY	VALUE
Product	XXXXX
Customer	XXXXX
Supplier	XXXXX
Location	XXXXX

- Highly fluid schema/no schema
- High performance for simple transactions
- Poor performance deep analytics

Graph Database



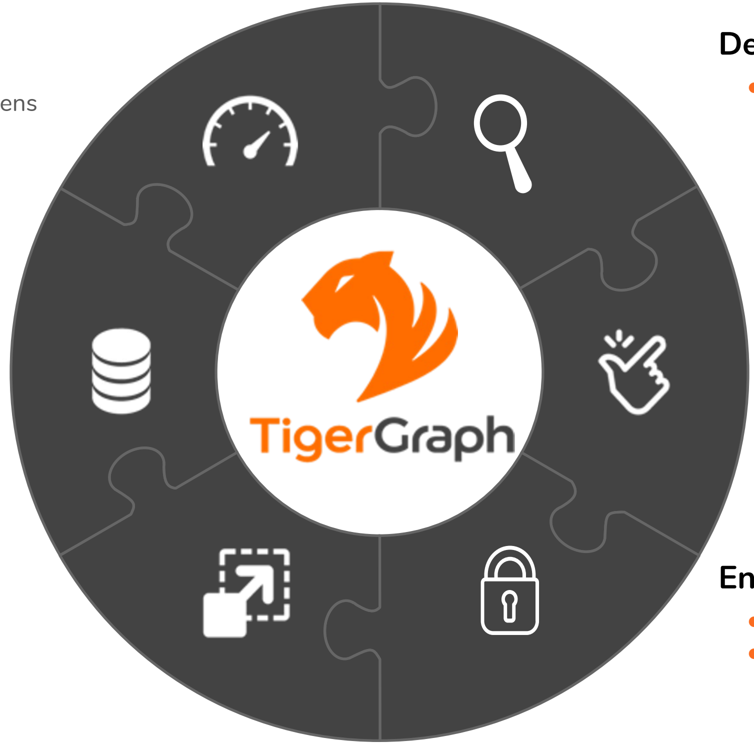
- Flexible schema
- High performance for complex transactions
- High performance for deep analytics

The Evolution of Graph Databases

	Graph 1.0 Single server, non-parallel	**Graph 2.0** NoSQL base for storage scale	**Graph 3.0** Native, Parallel
Native Graph Storage	✓	✗ Key-value or column store	✓
Parallel Loading	✗ Days to load terabytes	✗ Days to load terabytes	✓ Hours to load terabytes
Parallel Multi-Hop Analytics	✗ Times out after 2 hops	✗ Runs out of time/memory after 2 hops	✓ Sub-second across 10+ hops
Parallel Updates (in real-time)	✗ Batch updates	✗ Batch updates	✓ Mutable/ Transactional
Scale Out for Speed and Size	✗ Requires manual sharding of data and queries	✓ Sharded graphs, handles 1-2 hop queries	✓ Auto distributed data & analytics, for TBs data & 2B+ deep queries/day
Privacy for Sensitive Data	✗	✗	✓ MultiGraph

Benchmark at https://info.tigergraph.com/benchmark

TigerGraph: 3rd Generation Graph Database



Real-time Performance

Sub-second response for queries touching tens of millions of entities/relationships.

Transactional (Mutable) Graph

- Hybrid Transactional / Analytical Processing (HTAP): OLTP and OLAP.
- Hundreds of thousands of updates per second, billions of transactions per day.

Scalability for Massive Datasets

100 B+ entities, 1 T+ relationships.

Deep Link Multi-Hop Analytics

- Queries traverse 10+ hops deep into the graph performing complex calculations.

Ease of Development & Deployment

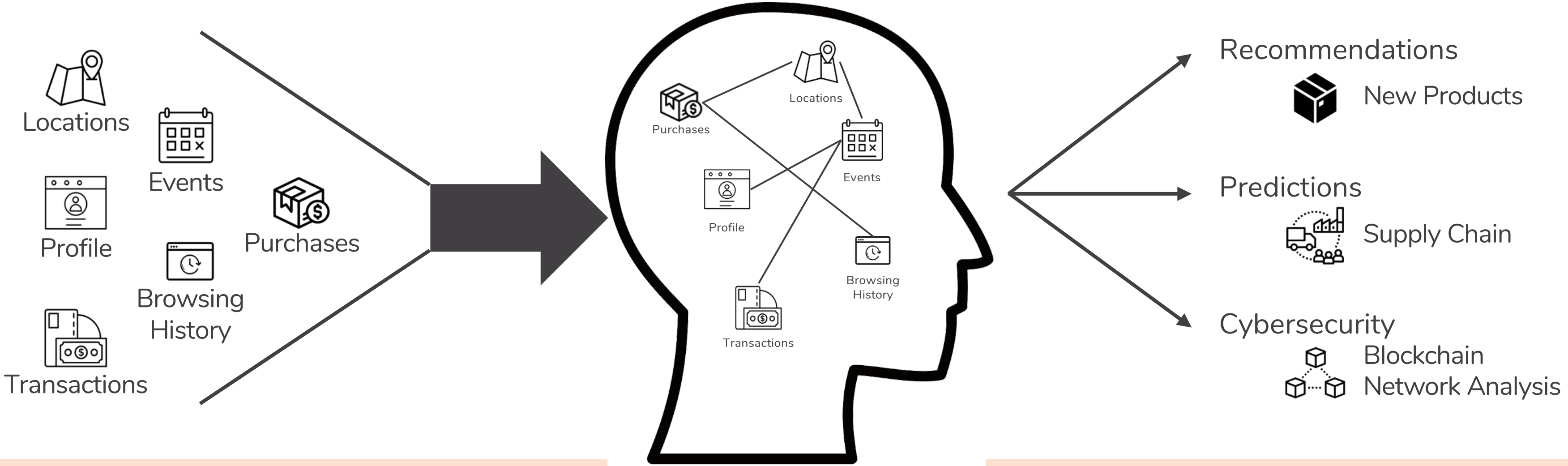
- GraphStudio - visual SDK.
- GSQL - Intuitive, Turing complete graph query language for developing complex analytics in days.
- User extensible graph algorithms library.

Enterprise Grade Security

- Encryption support.
- Control access to sensitive data based on user role, department or organization with MultiGraph.

Graph is How We Think

Graph is a natural model for interconnected data.
It is an organic way of modeling data for a variety of relationships and transactions.



Locations

Events

Profile

Purchases

Browsing History

Transactions

Recommendations

New Products

Predictions

Supply Chain

Cybersecurity

Blockchain

Network Analysis

Identify key data and process massive amounts of data

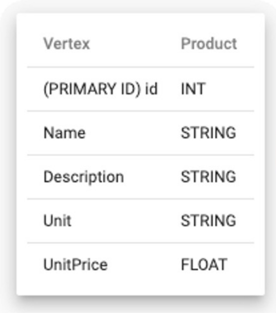
Use the power of relationships and deep analytics to provide insights

What is a graph?

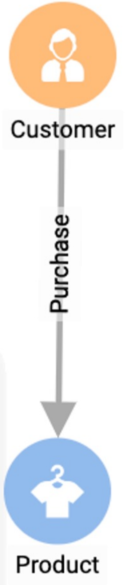
A collection of

- Vertices (vertexes, nodes) and
- Edges (links, relationships) between vertices

- A graph is an object consisting of two sets called vertex set and its edge set.
- The vertex set is a finite nonempty set.
- The edge set may be empty, but otherwise its elements are two-element subsets of the vertex set.
- The elements of the vertex set of a graph are called vertices and the elements of the edge set are called edges.



Vertex	Product
(PRIMARY ID) id	INT
Name	STRING
Description	STRING
Unit	STRING
UnitPrice	FLOAT



What is a vertex? What is an edge?

- Vertices represent real life entities
  - People, products, parts, IP addresses, genre, location, date, ...
  - Visualised as circle/disc
  - Can be considered similar to a relational database table, but ...
  - Can be called nodes, but could that be unambiguous

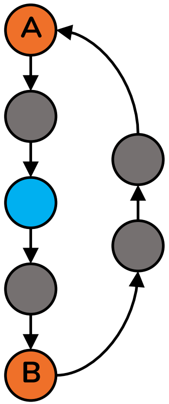
- Edges represent relationships between real life entities
  - Parenthood, friendship, ownership, hierarchy, action, ...
  - Visualised as line or arrow
  - Can be considered similar to a join in a relational database, but ...
  - Sometimes a relationship can be better represented as a vertex

Vertices and Edges

- Vertices and edges
  - Have type
  - Can have attributes
- Vertices have (primary) IDs
  - Unique value within a vertex type
  - If you insert a vertex with the same ID, it will overwrite the previous one
  - Similar to primary key in relational databases
- Edges do not have IDs
  - They have source and target vertices (vertex IDs)
  - Their combination identifies the edge uniquely

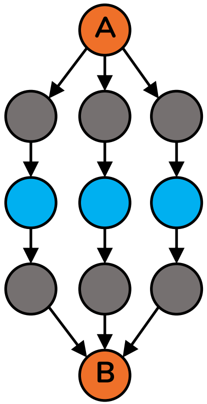
Seven Key Data Science Capabilities

1 Deep Link Analysis



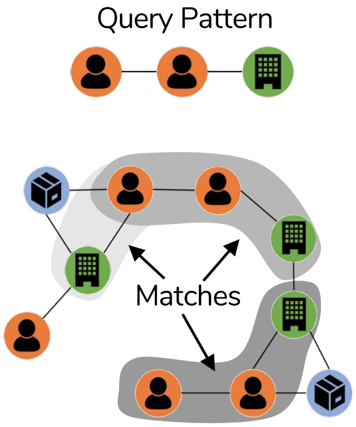
For a set of entities (e.g., customers, accounts, citizens, doctors), show all links or connections

2 Subgraph or Relationship Discovery & Computation



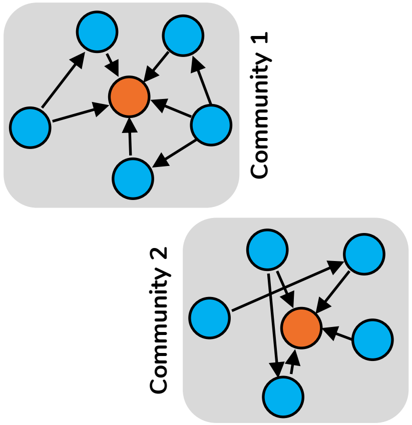
Given a graph (e.g., all payments or claims for a set of customers), find all the relevant subgraphs

3 Multi-dimensional Entity & Pattern Matching



Given a graph (e.g., all payments or claims for a set of customers), find similar graphs

4 Hub & Community Detection



Find most influential members of a group (customers, doctors, citizens) & detect community around them

5 Geospatial graph analysis — Analyze changes in entities & relationships with location data

6 Temporal (time-series) graph analysis — Analyze changes in entities & relationships over time

7 ML feature generation and explainable AI — Extract graph-based features to feed as training data for ML; power explainable AI

Community resources

Developer Community

TG Community Forum	community.tigergraph.com
TG Community Chat	discord.gg/F2c9b9v
Developer Portal	https://dev.tigergraph.com/
Reddit	reddit.com/r/tigergraph/
YouTube	youtube.com/tigergraph
LinkedIn	linkedin.com/company/tigergraph/
Twitter	twitter.com/tigergraphdb
Twitch	twitch.tv/tigergraph
GitHub	github.com/tigergraph/ecosys github.com/tigergraph/gsql-graph-algorithms github.com/TigerGraph-OSS

Thank you!

Szilard Barany
szilard.barany@tigergraph.com
Discord: Szilard 🐅