ORACLE

Machine Learning with Oracle Graph

Natwest Developer General Session

March 2021

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

> “Our mission is to become the trusted advisors for Developers through inspiration and engagement”

DevRel Team

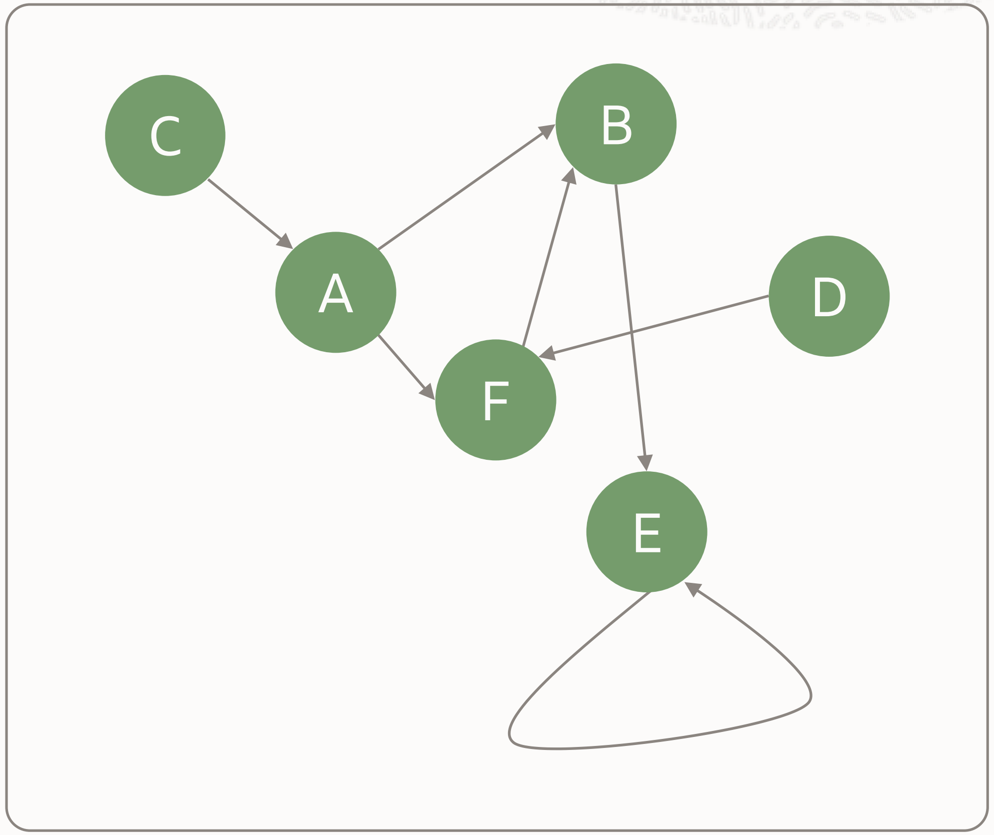
Graph data model

What is a graph?

- Data model representing entities as vertices and relationships as edges
- Optionally including attributes

What are typical graphs?

- Social networks
  - LinkedIn, Facebook, Twitter, …
- Physical networks, supplier networks, …
- Dependency graphs
  - Part hierarchies, data lineage, org charts, …
- Workflows
- Knowledge graphs
  - Associating related entities, eg. in public safety



Why are graphs so popular?

Easy data modeling

- All you need is pen and paper

Flexible data model

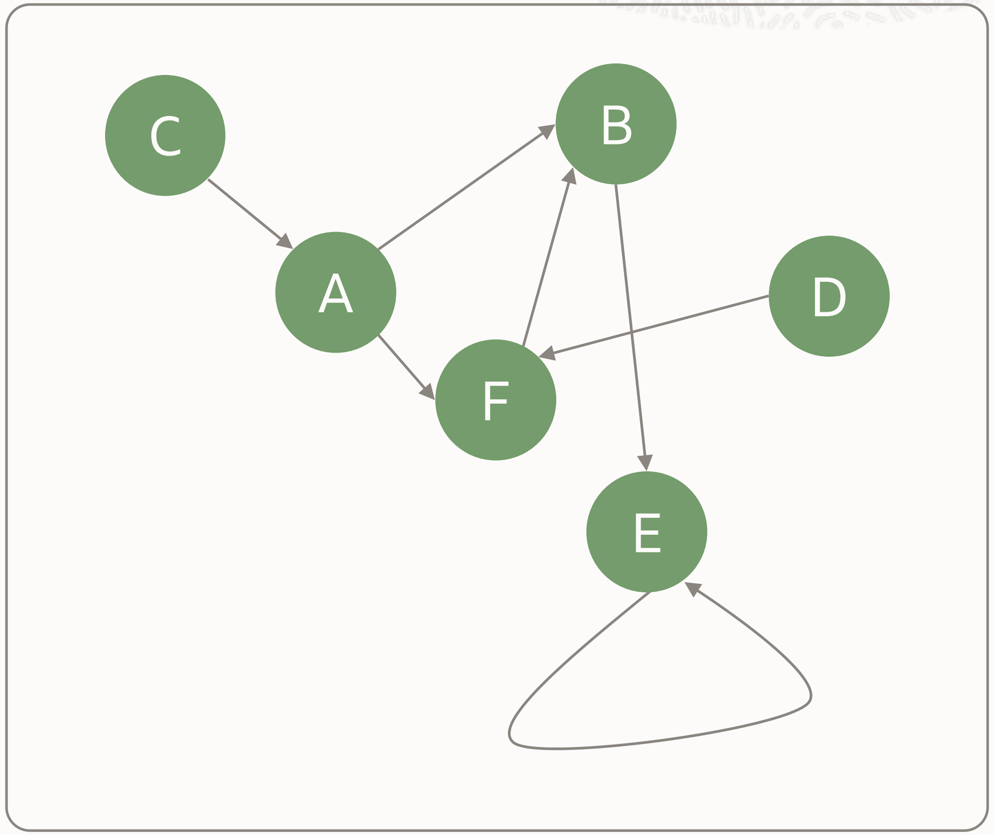
- No predefined schema, easily extensible
- Particularly useful for sparse data

Insight from graphical representation

- Intuitive visualization

Enable new kinds of analysis

- Graphs store rather than compute relationships so traversals (friend-of-a-friend, is there a path) are quick
- Additional features for Machine Learning

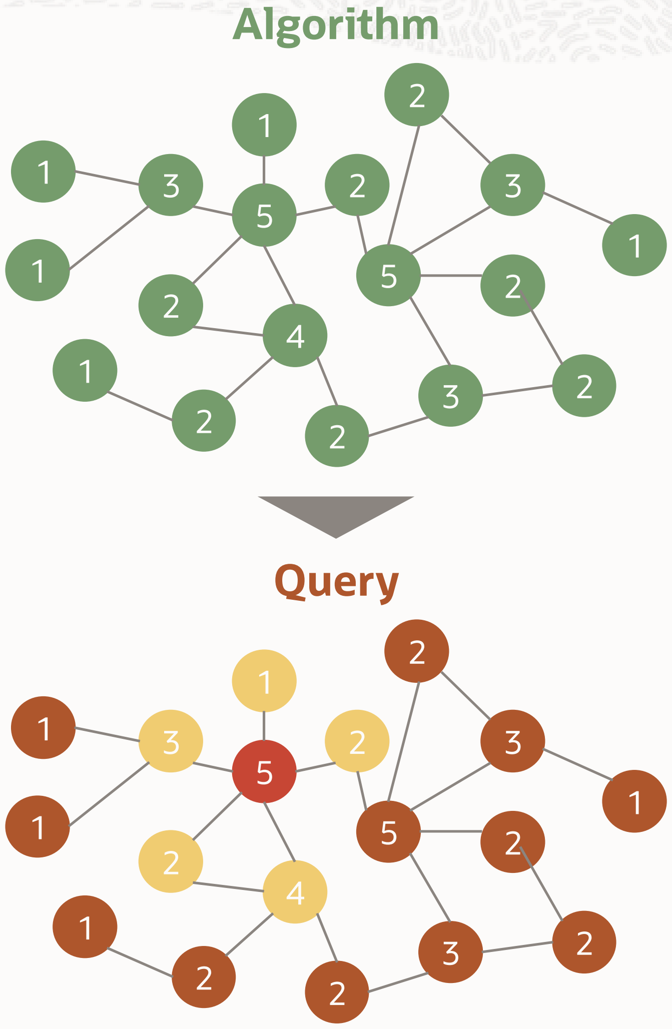


Two types of processing in graph analytics

Oracle’s Graph Engine can execute both queries and algorithms, and combinations of the two

- Query (PGQL)
- Algorithm
- Mutation

Which node up to 3 hops from a given node has the highest importance (e.g. degree centrality, page rank, betweenness centrality, …)?



Pattern matching–PGQL queries

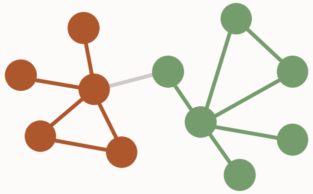
PGQL MATCH clause describes graph pattern

```
SELECT p.name
FROM MATCH (m:Person)-[:likes]->(p:Person)
WHERE m.name = 'Mario'
```

Find common ancestors of Mario and Luigi

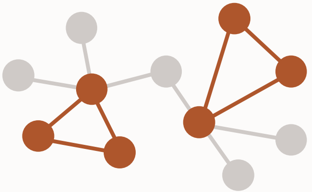
```
PATH has_parent AS () -[:has_father|has_mother]-> (:Person)
SELECT ancestor.name
FROM MATCH (p1:Person) -/:has_parent+/-> (ancestor)
   , MATCH (p2:Person) -/:has_parent+/-> (ancestor)
WHERE p1.name = 'Mario’
  AND p2.name = 'Luigi'
```

Graph analytics: 60+ parallelized, in-memory algorithms out-of-the-box



Detecting components and communities

Strongly Connected Components,
Weakly Connected Components,
Label Propagation,
Conductance Minimization,
Infomap

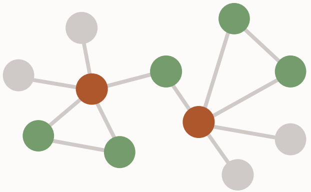


Evaluating structures

Adamic-Adar Index, Conductance,
Cycle Detection, Degree Distribution,
Eccentricity, K-Core, LCC, Modularity,
Reachability Topological Ordering,
Triangle Counting

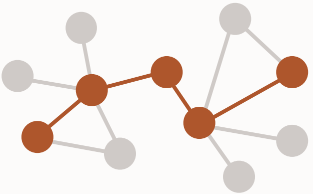
Link prediction and others

WTF (Who to follow)
Minimum Spanning-Tree,
Matrix Factorization



Ranking and walking

PageRank, Personalized PageRank,
Degree Centrality, Closeness Centrality,
Vertex Betweenness Centrality,
Eigenvector Centrality, HITS, SALSA,
Random Walk with Restart



Path-finding

Shortest Path (Bellman-Ford, Dijkstra,
Bidirectional Dijkstra), Fattest Path,
Compute Distance Index,
Enumerate Simple Paths,
Fast Path Finding, Hop Distance

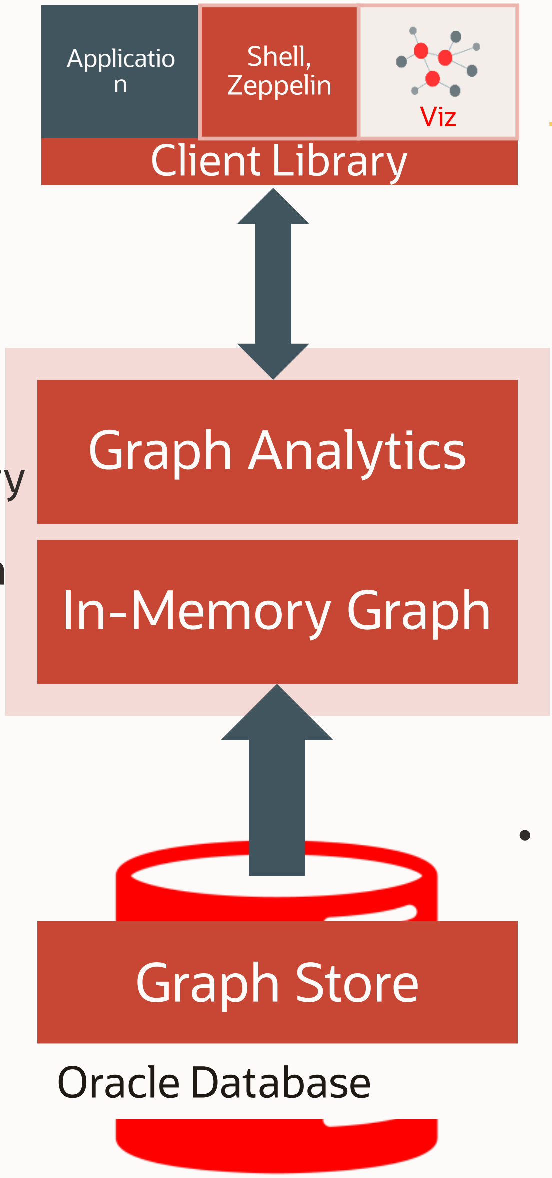
Machine learning

DeepWalk *
Supervised GraphWise *
Pg2Vec *

*on-premises, not yet in Graph Studio

Oracle Database Spatial and Graph

- In-memory parallel graph analytics server (PGX)
  - Load graph into memory for analysis
  - Automate graph refresh



Applicatio n

Shell, Zeppelin

Viz

Client Library

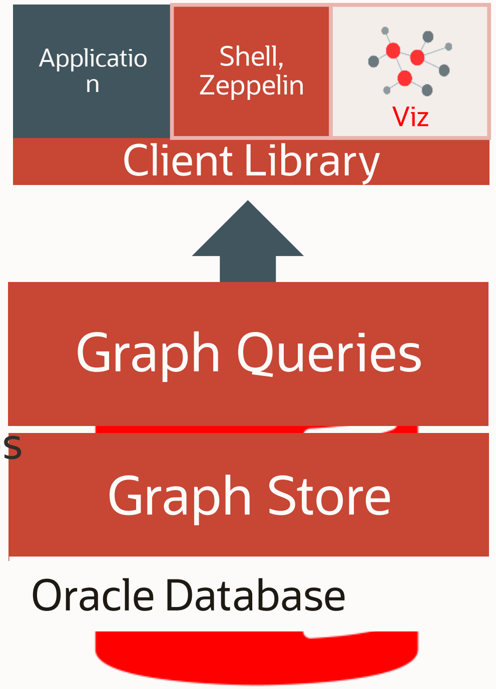
Graph Analytics

In-Memory Graph

Graph Store

Oracle Database

- Client libraries
  - Java API to develop applications
  - Command-line submission of graph queries
  - Graph visualization tool
  - APIs to update graph store



Applicatio n

Shell, Zeppelin

Viz

Client Library

Graph Queries

Graph Store

Oracle Database

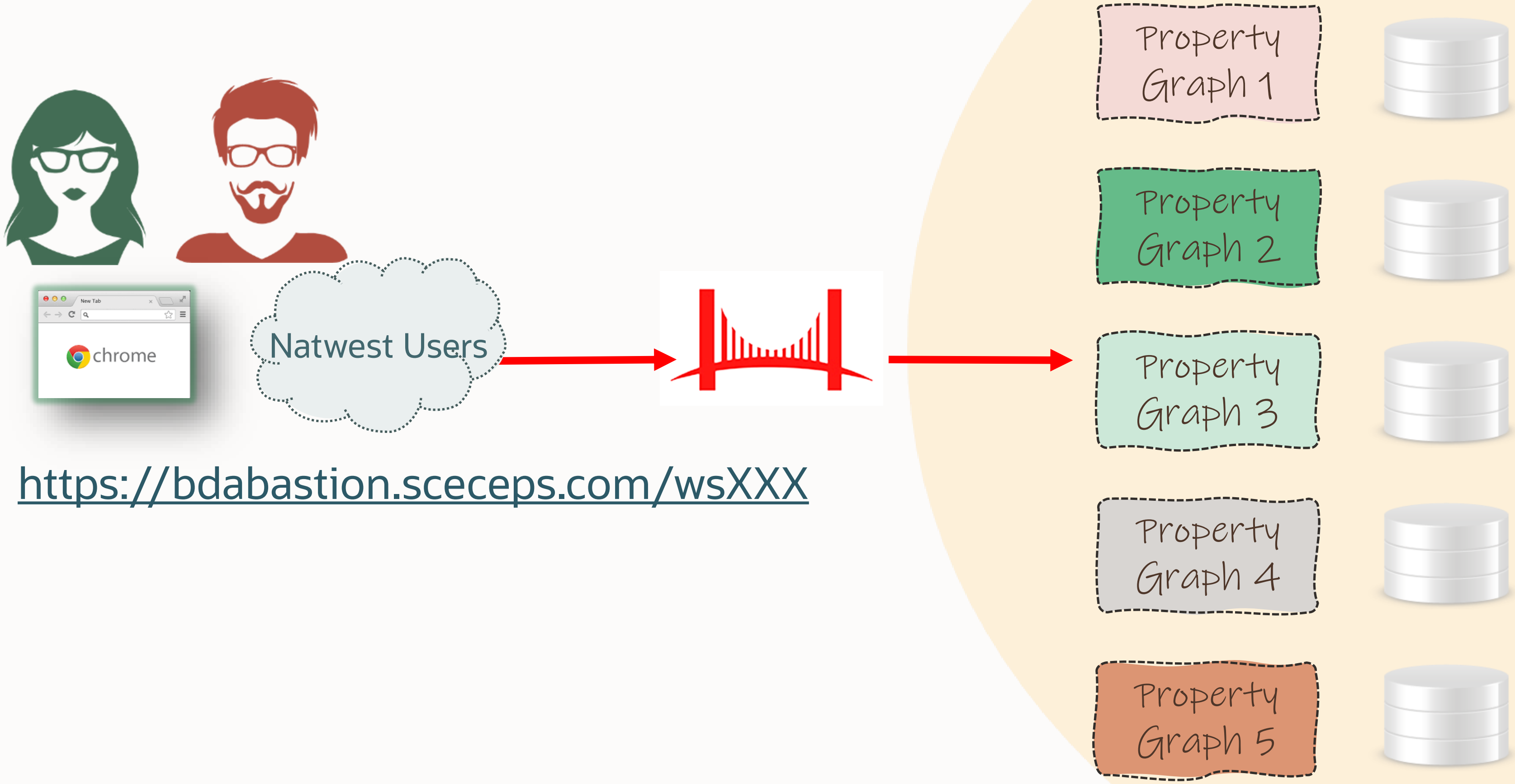
- In-database parallel graph traversal
  - Run PGQL queries in the database

Interaction with the Property Graph

- Access through APIs
  - Implementation of Apache Tinkerpop Blueprints APIs
  - Based on Java, REST plus SolR Cloud/Lucene support for text search
- Scripting
  - Groovy, Python, JavaScript, …
  - Apache Zeppelin integration, JavaScript (node.js) language binding
- Graphical UIs
  - Cytoscape, plug-in available
  - Commercial tools such as TomSawyer Perspectives
  - Vis.js and D3 among others

Labs Environment

ORACLE Cloud Infrastructure



https://bdabastion.sceceps.com/wsXXX

Materials & Documentation https://github.com/operard/mlgraph

Workshop / Hands-on Labs
Breakout Rooms

Room 1

Olivier Perard

Data Scientist Advocate

Room 2

Andrea Marchesini

Head of Innovation and
Educational Programs

Room 3

Francisco Alvarez

Principal Technology Architect

Room 4

Ismail Syed

Cloud Specialist Engineer

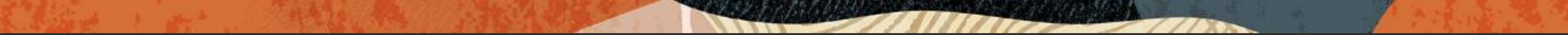
Room 5

Eric Steele

OCI PM EMEA

Thank You

DevRel Team



ORACLE