



Break New Ground

San Francisco

September 16–19, 2019

Using Graph Analytics for New Insights

Melli Annamalai

Senior Principal Product Manager

Oracle

September 17, 2019

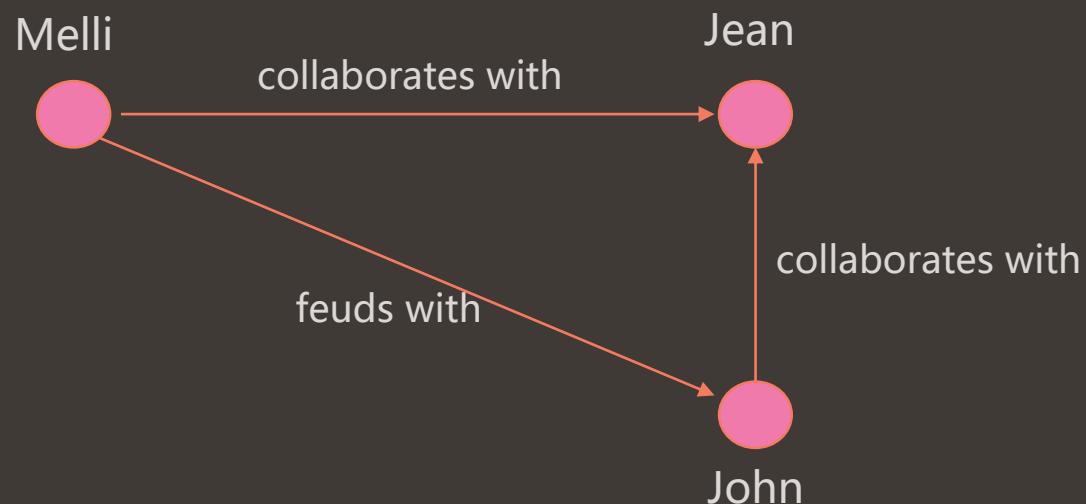
Safe Harbor

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.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

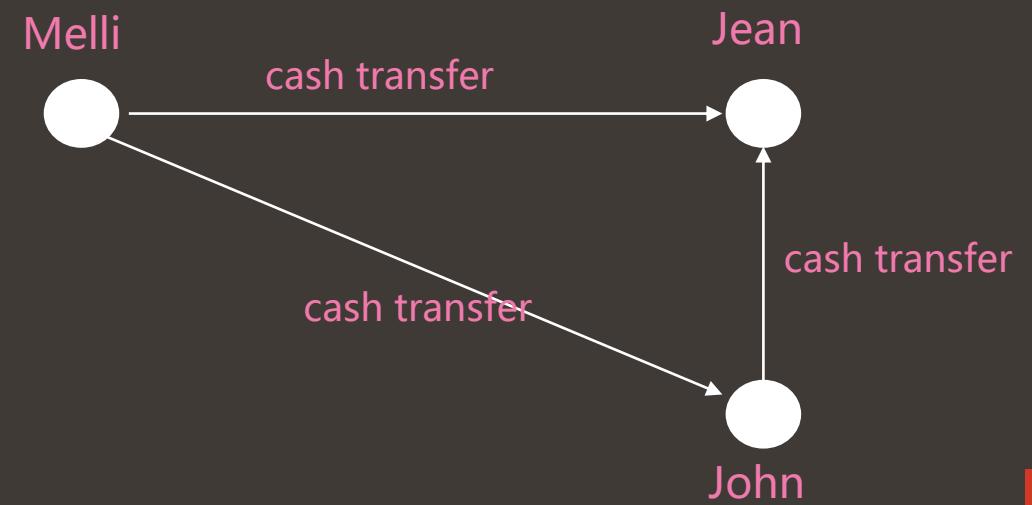
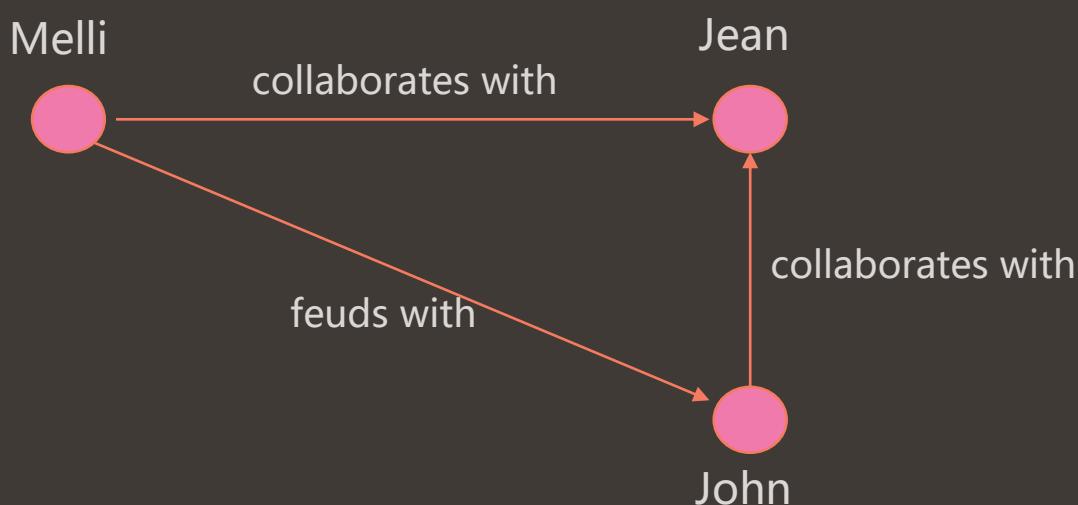
What is a Graph?

A collection of points (vertices) and lines between those points (edges)



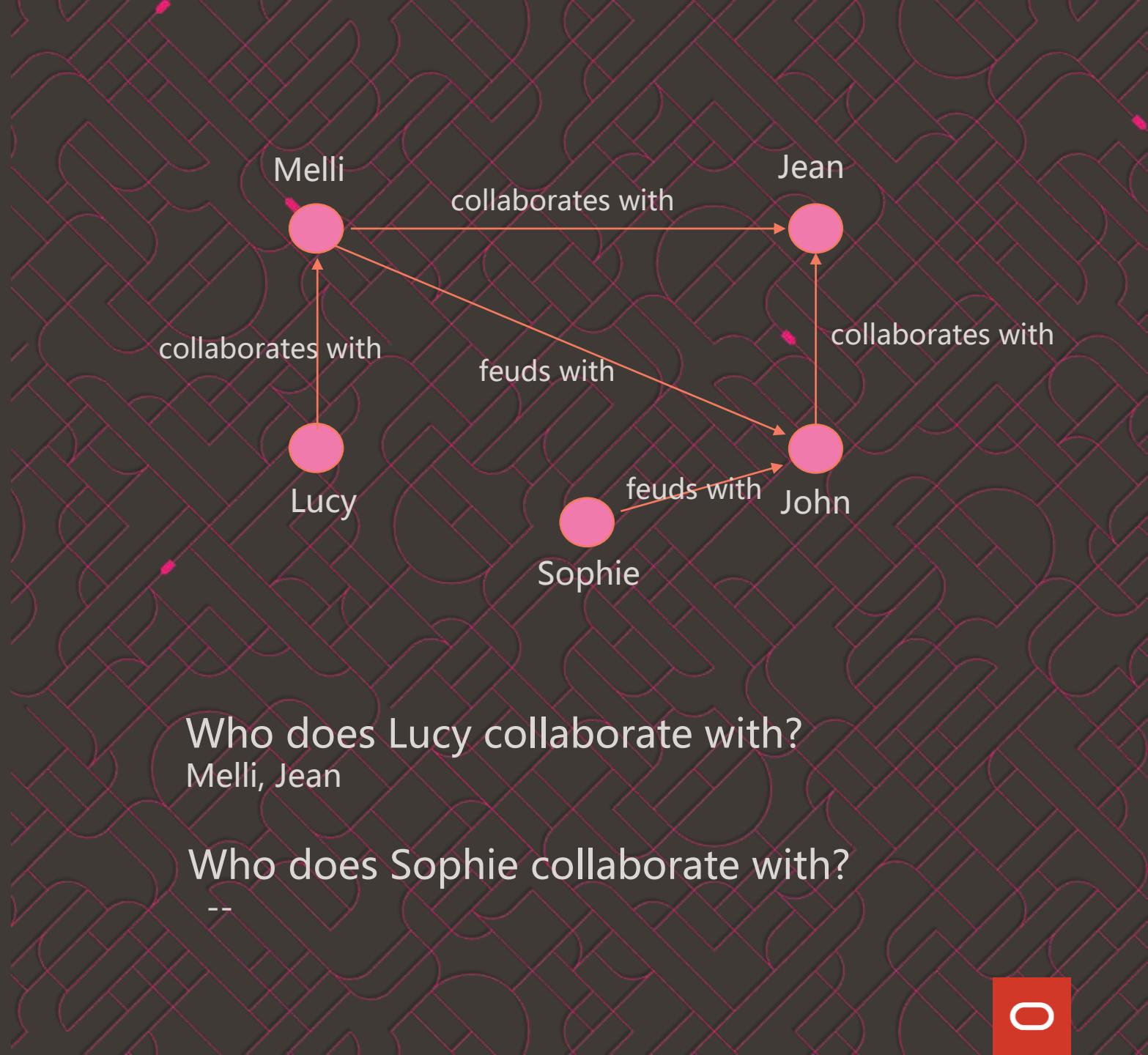
What is a Graph?

A collection of points (vertices) and lines between those points (edges)



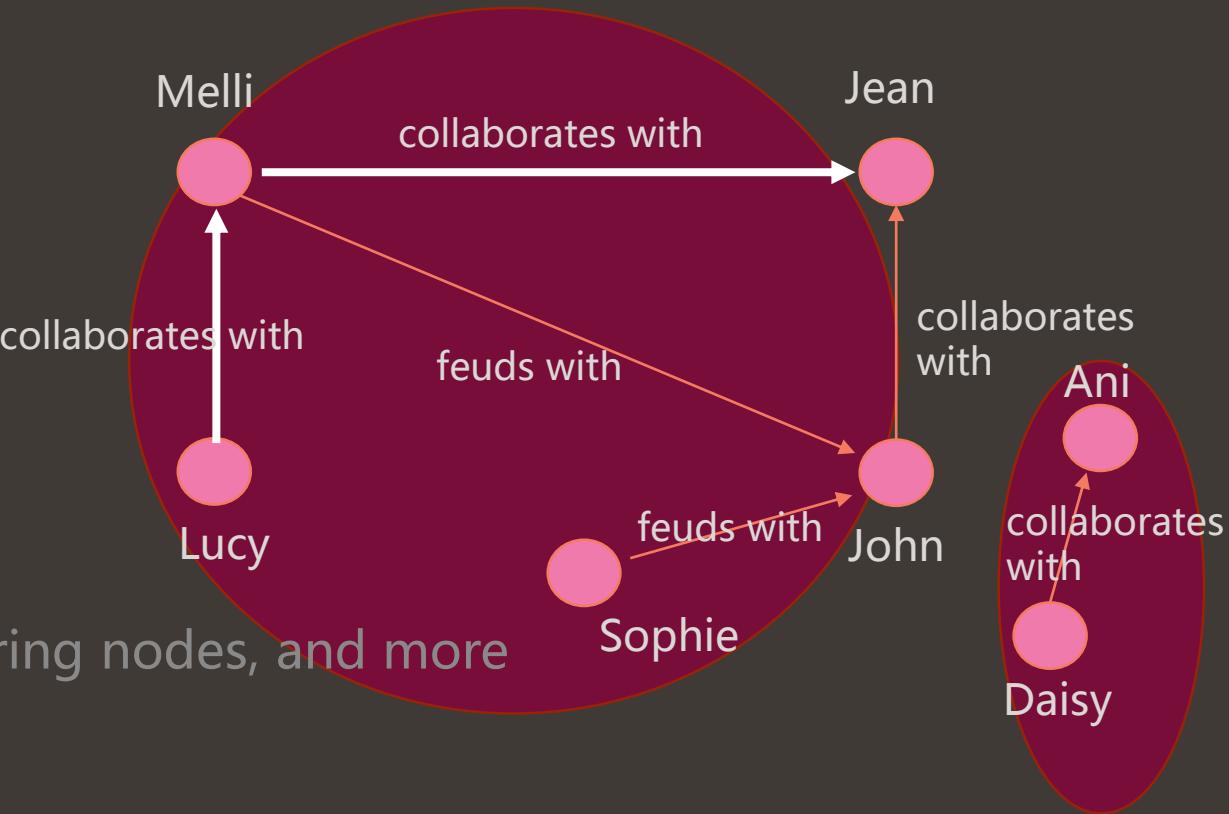
Why are Graphs important?

- Captures relationships between data entities
- Use relationships (connections) in data analysis

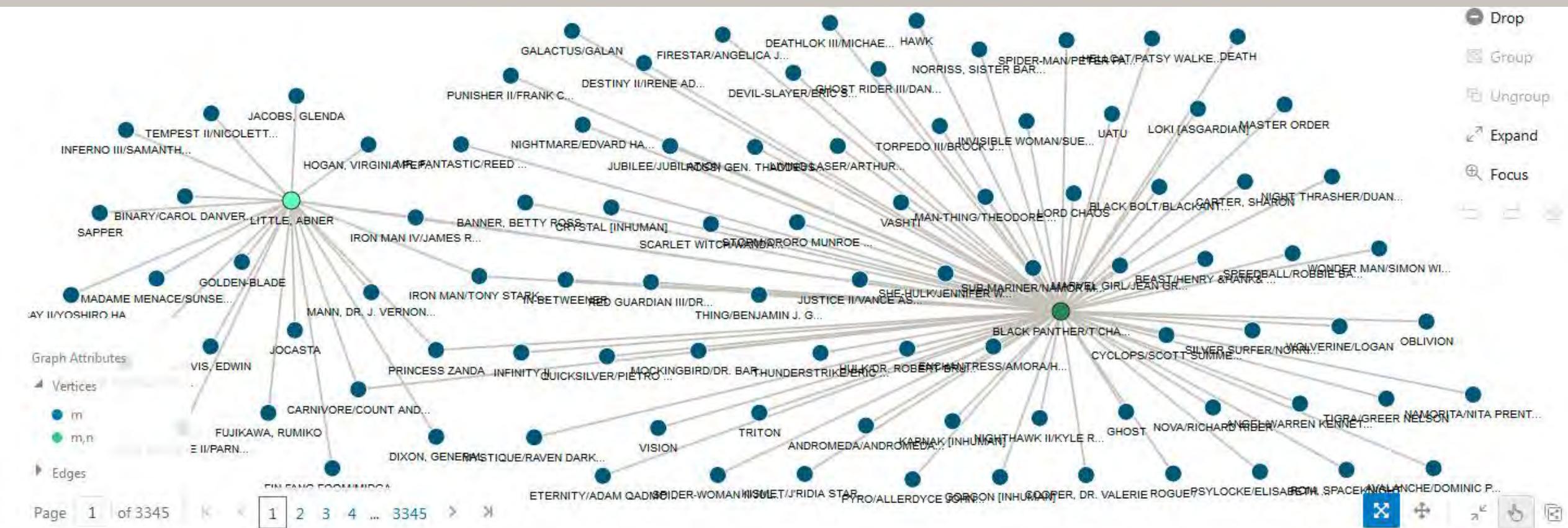


What Can We Do With Graphs?

- Path between entities
- Distance between entities
- Importance of entities
 - By incoming edges, importance of neighboring nodes, and more
- Clustering of entities
- And more ...

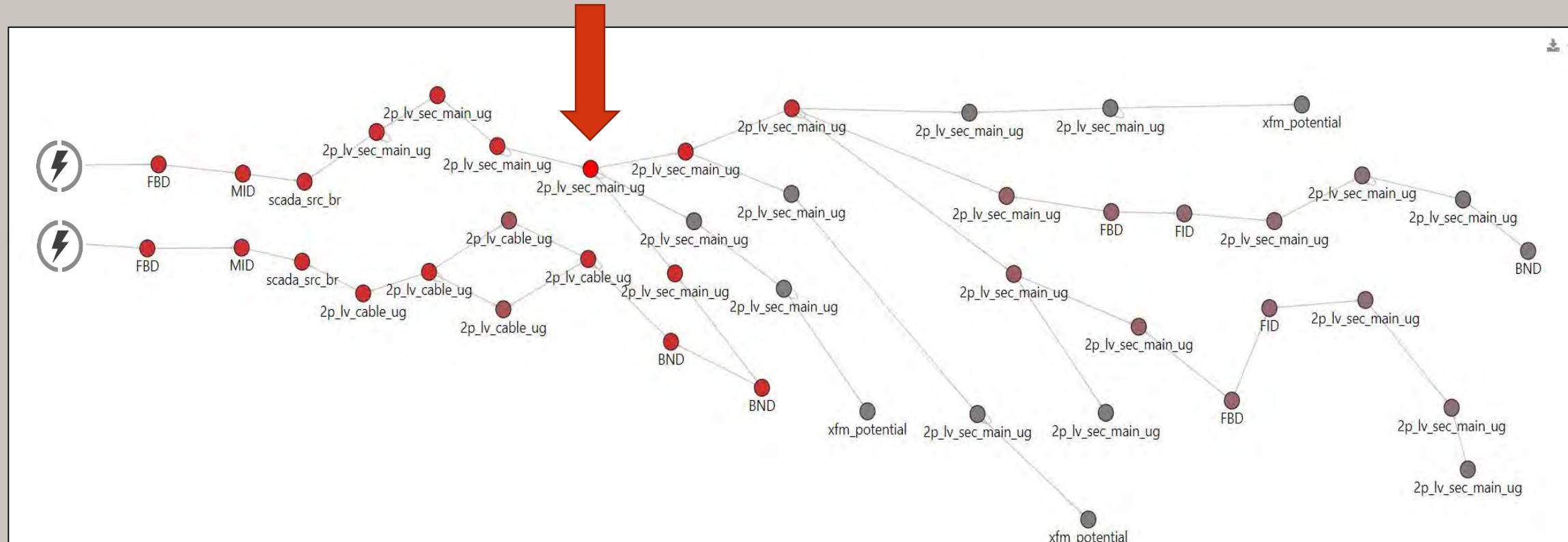


Find popular nodes in a social network



Identify influencers

Find nodes that are in most shortest paths
between other nodes



Identify nodes that can cause vulnerabilities in a grid

Topics

Data Analytics &
Data Science



Andrew Ross

18 February 2019

Gartner: top 10 data and analytics technology trends for 2019

Trend #5: Graph

According to Gartner, graph analytics is a set of analytic techniques that help enterprises explore the relationships between entities of interest such as transactions, processes and staff.

The application of graph processing and graph database management systems will grow at 100% annually through 2022.

Gartner MQ - Data Management Solutions for Analytics

Figure 1. Magic Quadrant for Data Management Solutions for Analytics



Graph Data Models

Property Graph Model

- Path Analytics
- Graph Analytics
- Detect patterns and anomalies

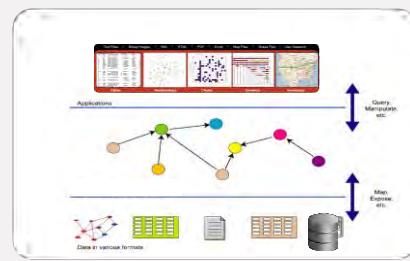


- Financial
- Retail, Marketing
- Social Media
- Smart Manufacturing

Shipping for 3+ years

RDF Graph Model

- Data federation
- Knowledge representation
- Semantic Web



- Life Sciences
- Health Care
- Publishing
- Finance

Graph Model

Use Cases

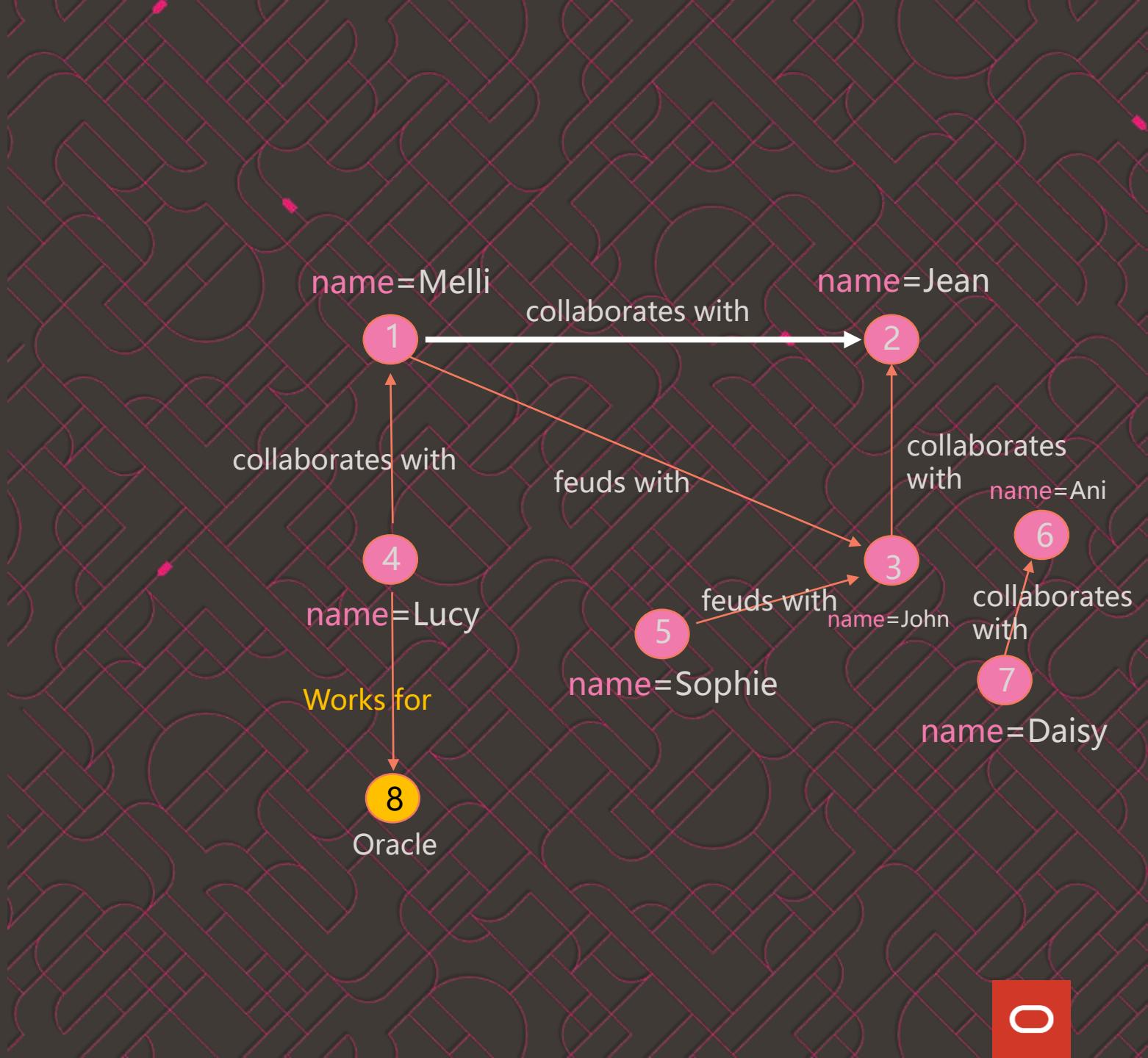
Industry Domain

Shipping for 12+ years

Property Graphs

Property Graph Model

- Edges/vertices have unique identifiers
- Edges/vertices have a collection or properties
- Vertices are of different kinds
- Edges have labels denoting type of relationship



Property Graph Product Overview

- Store, manage, query and analyze graphs
- Highly scalable in-memory analytics

10s of billions of edges and vertices

- 50+ pre-built graph analysis algorithms

Detecting components and communities

Tarjan's, Kosaraju's, Weakly connected components, label propagation, etc.

Ranking and walking

Pagerank, personalized pagerank, betweenness centrality, etc.

Evaluating community structures

Conductance, modularity, triangle counting, Adamic-Adar, etc.

Path-finding

Path distance, Dijkstra's, Bellman-Ford's, etc.

Property Graph Product Overview

- **PGQL: Powerful graph query language**

SQL-like language for specifying graph patterns
Working on graph additions to the SQL standard

PGQL example:

```
PATH any_edge as ()-[]-()
SELECT n, m MATCH(n) -/: any_edge/ -> m
WHERE n.name='Lucy' and m.name='Jean'
```

- Java API for analytics
- Rich user interface

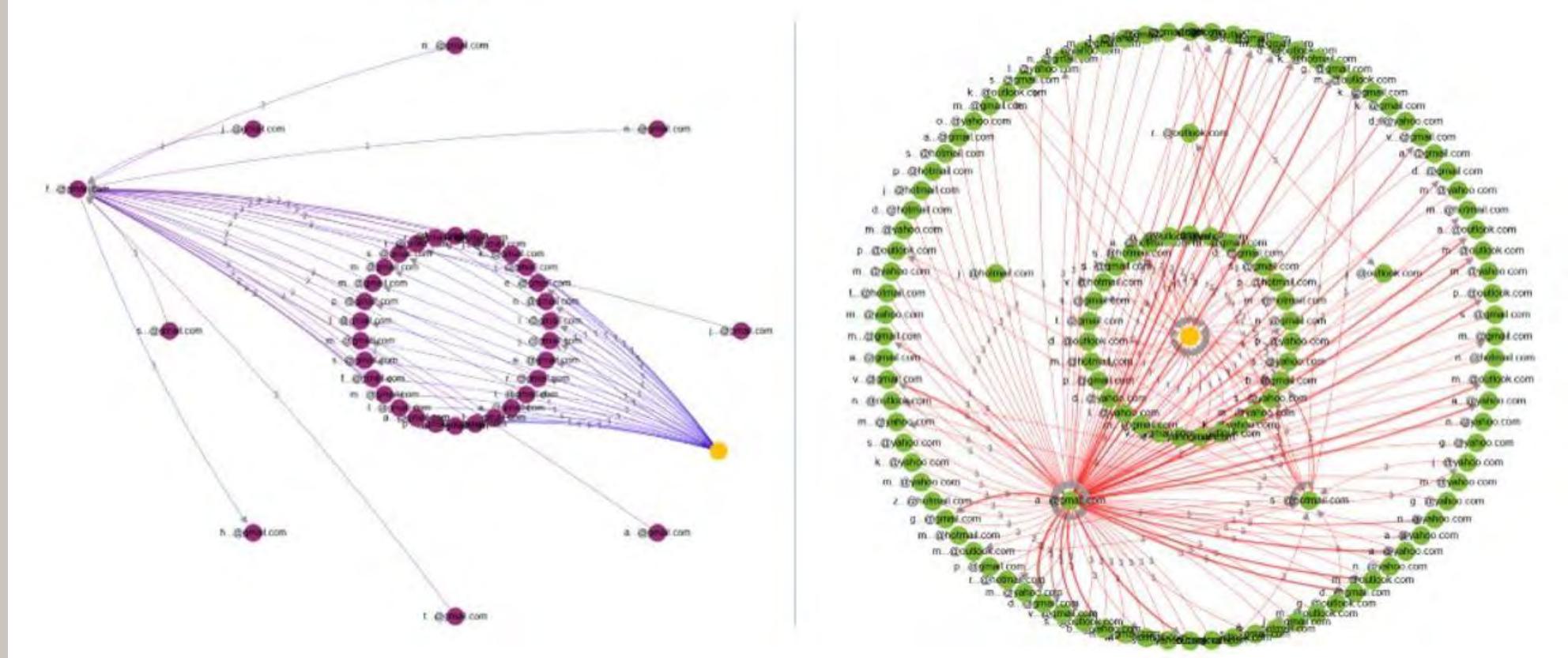
Notebook
Shell UI
Graph Visualization

- **Enterprise capabilities – built on Oracle infrastructure**

Manageability, fine-grained security, high availability, integration, and more

Example Insights from Graph Analytics

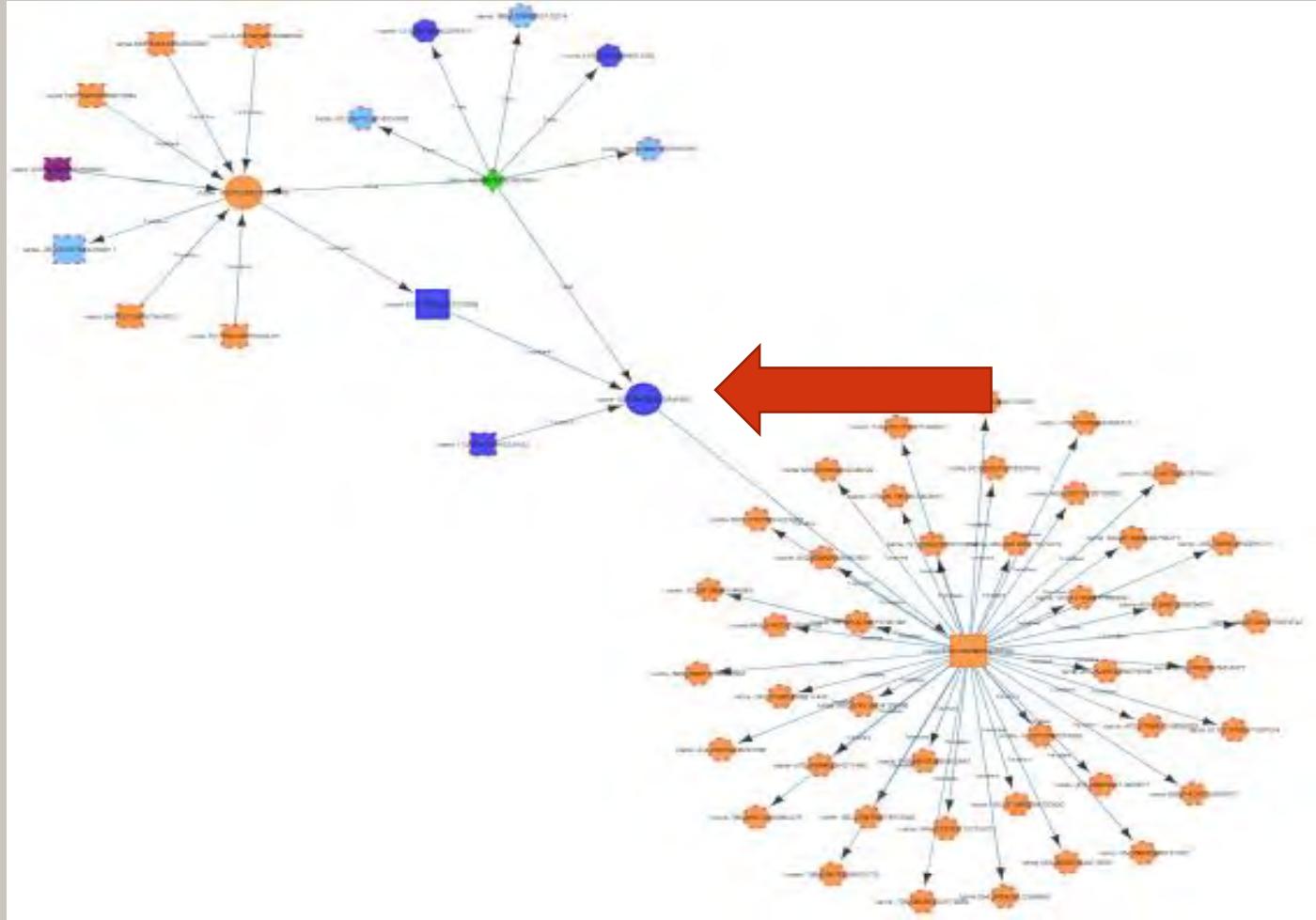
Paysafe: Money Transfer Fraud Detection



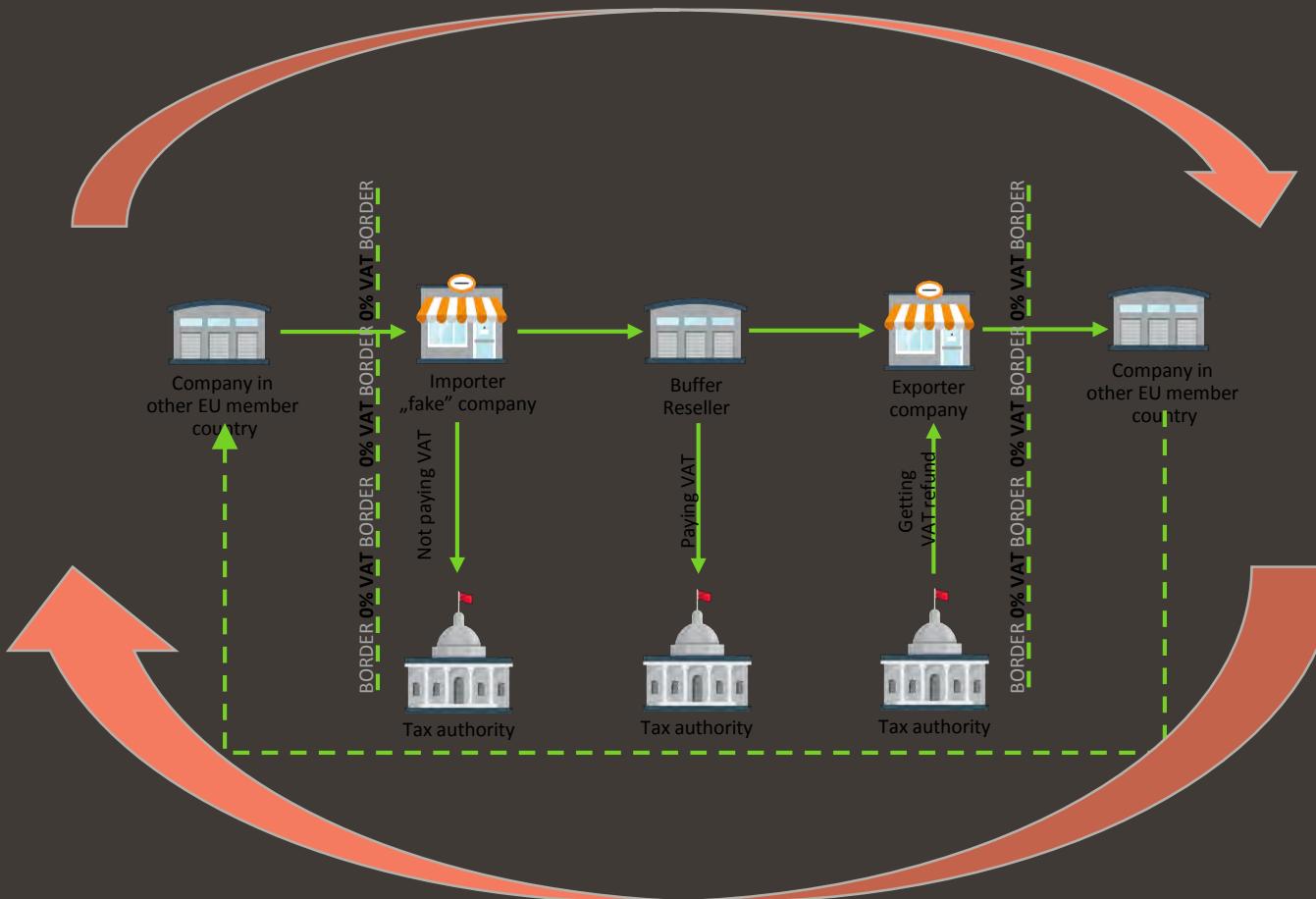
Multiple paths going to the same destination

Limited number of source and destination vertices

Banco De Galicia: Which Nodes are Transferring money to other Banks?

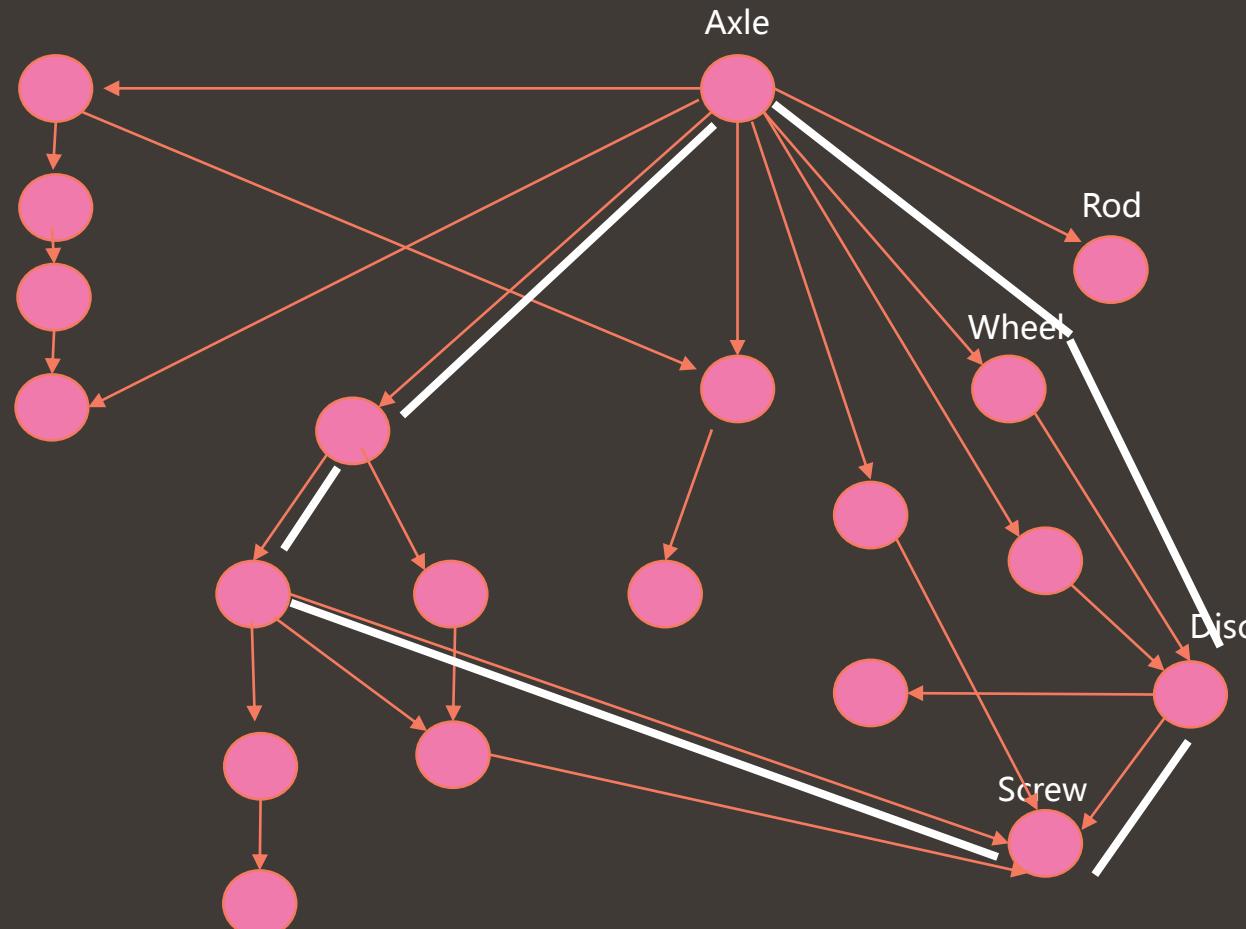


Circular Payments and Tax Fraud



Manufacturing: What is the Impact of Changing this Part?

A car has 30,000 parts



Property Graph Deep Dive

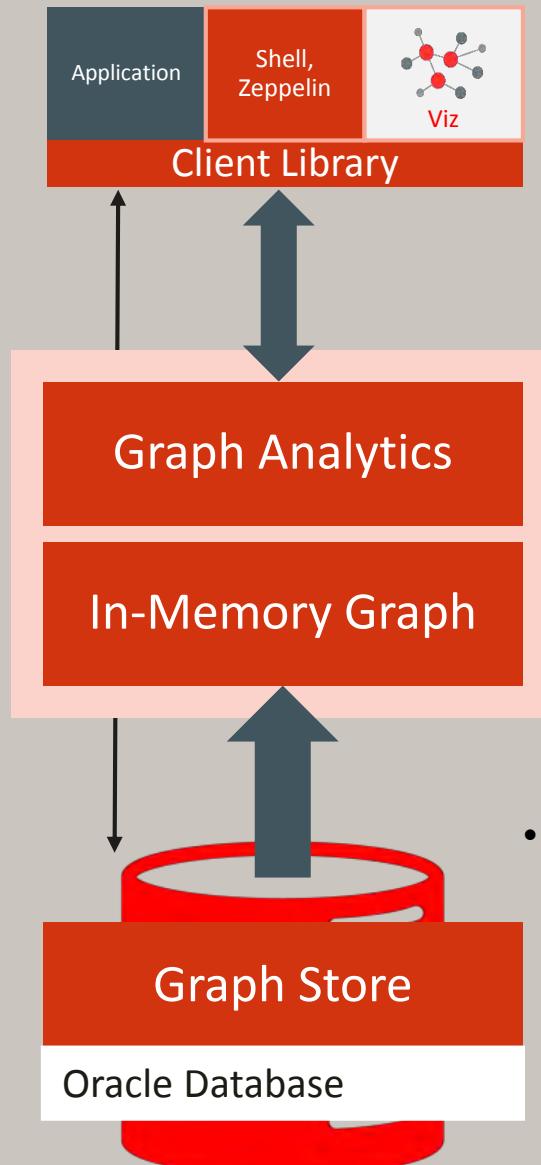
Graph Components

- PGQL: SQL-like Graph query language
- PGX: In-memory analytics server
- Graph Storage

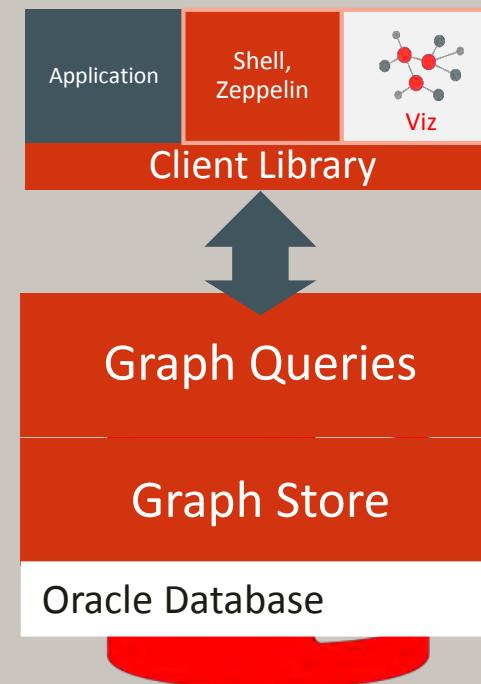
Architecture

Product:
Oracle Database
Spatial and Graph

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



- Client libraries
 - Java API to develop applications
 - Command-line submission of graph queries
 - Graph visualization tool
 - APIs to update graph store
- In-database parallel graph traversal
 - Run PGQL queries (converted to SQL) in the database

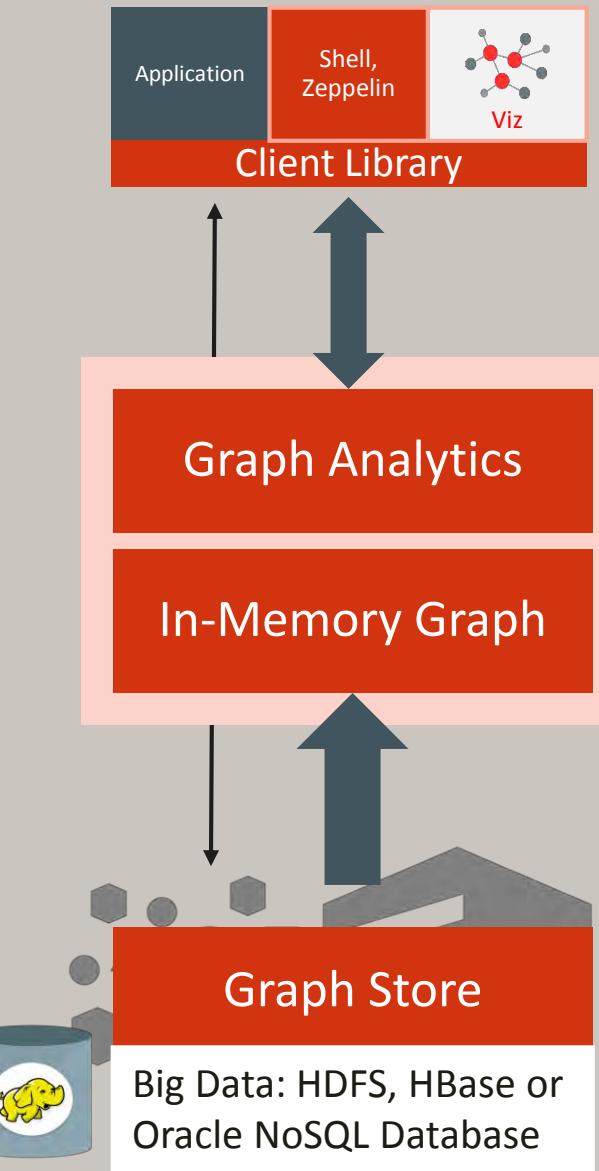


Using Graphs on a Big Data Platform

Product: Oracle Big Data Spatial and Graph

Runs on the Hadoop platform

- In-memory parallel graph analytics server (PGX)
 - Load graph into memory for analysis
 - Automate graph refresh
- Client libraries
 - Java API to develop applications
 - Command-line submission of graph queries
 - Graph visualization tool
 - APIs to update graph store



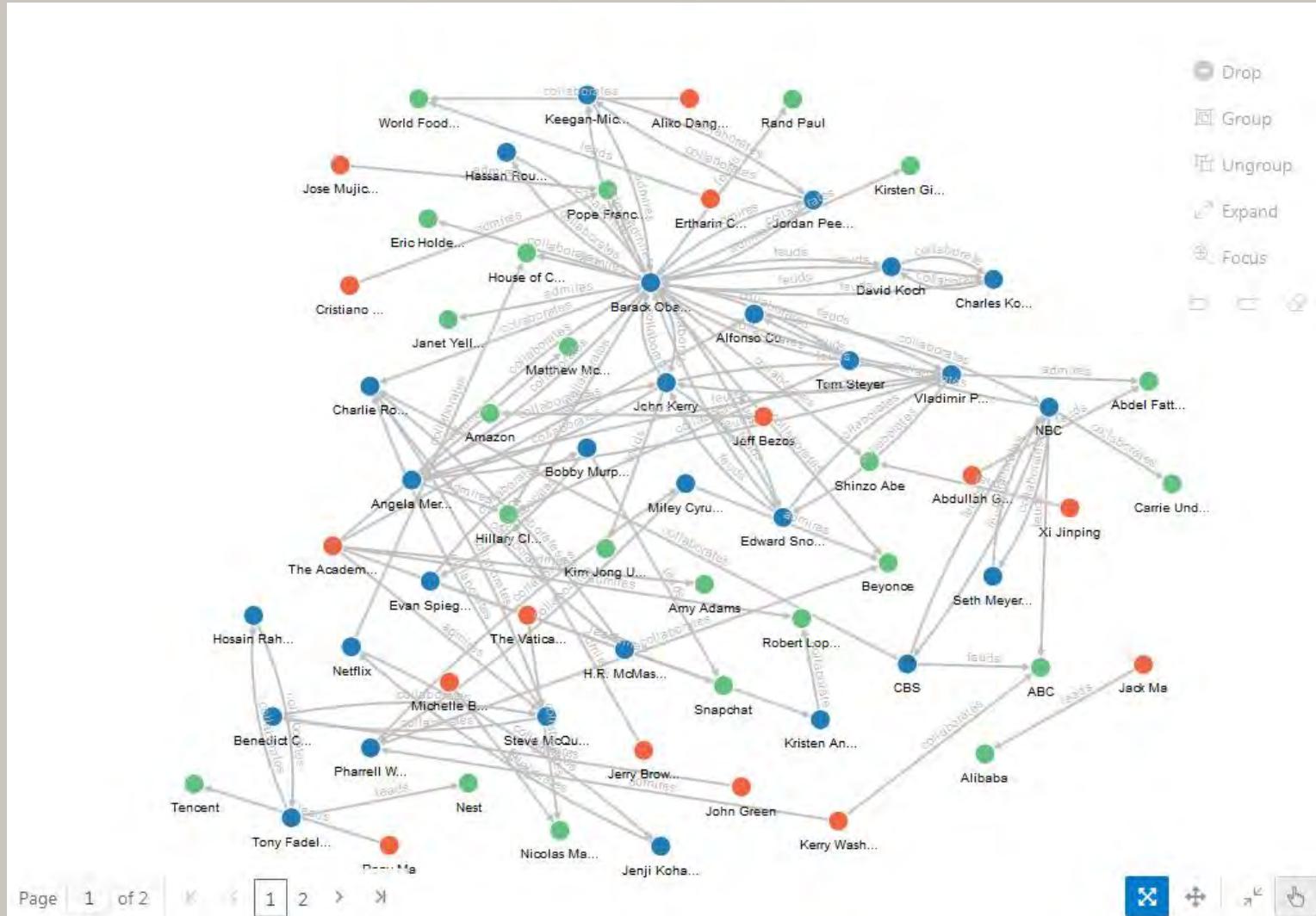


Load Graph into Memory for Analytics

Load Graph

- From Oracle Database
- From Oracle NoSQL Database
- From Apache HBase
- From files

‘Connections’ Graph



‘Connections’ dataset in tables

Organizations

org_id	name	type	religion	genre	country
	CBS	TV Netw			
	Amazon	Company			

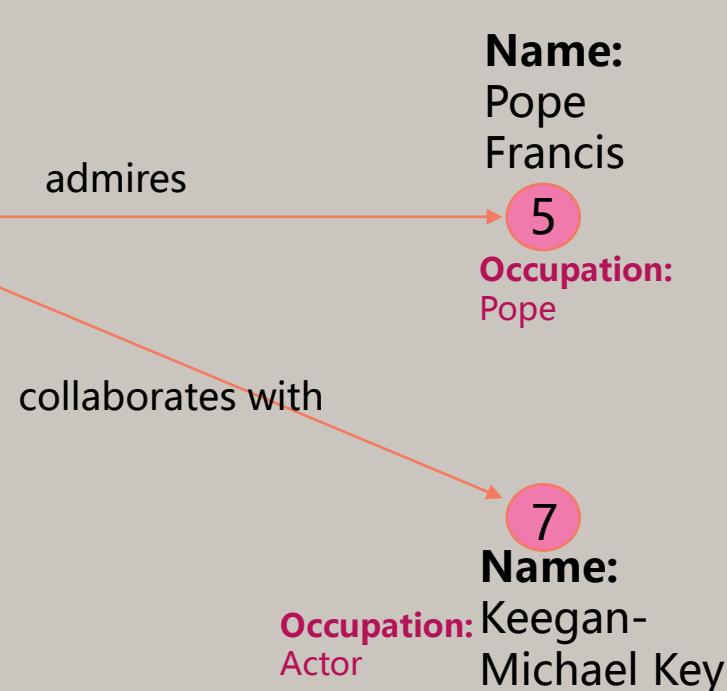
Relationships

relation_id	from_id	from_type	to_id	to_type	relation_type
	1	person	5	person	admires
	1	person	7	person	collaborates

People

person_id	name	company	occup-ation	country
5	Pope Francis		Pope		Vatican
1	Barack Obama		44 th Pres.		USA
7	Keegan-Michael Key		Actor		USA

Name:
Barack Obama
1
Occupation:
Former President



Storing the Graph

Vertex table

Column	Data Type	Usage
VID	NUMBER	Unique vertex id
K	NVARCHAR2(3100)	Property name
T	INTEGER	Data type
V	NVARCHAR2(15000)	String value
VN	NUMBER	Numeric value
VT	TIMESTAMP(6) WITH TIMEZONE	Date value



Edge table

Column	Data Type	Usage
EID	NUMBER	Unique edge id
SVID	NUMBER	Id of origin vertex
DVID	NUMBER	Id of destination vertex
EL	NVARCHAR2(3100)	Edge label
K	NVARCHAR2(3100)	Property name
T	INTEGER	Data type
V	NVARCHAR2(15000)	String value
VN	NUMBER	Numeric value
VT	TIMESTAMP(6) WITH TIMEZONE	Date value

Columnar representation of vertex and edge tables

.ope, .opv
(Oracle file format)

movies.opv
1,name,1,Tom%20Hanks,, 1,age,2,,61, 1,label,1,actor,, 2,name,1,Meg%20Ryan,, 2,age,2,,55 2,label,1,actress,, 3,title,You've%20Got%20Mail,, 3,year,2,,1998, 3,label,1,movie,,

movies.ope
7,1,3,actedIn,weight,4,,1.0, 8,2,3,actedIn,weight,4,,1.0,

EDGE_LIST

```
"IRON MAN IV/JAMES R.", "FORTUNE", "DOMINIC"  
"IRON MAN IV/JAMES R.", "ERWIN", "CLYTEMNESTRA"  
"IRON MAN IV/JAMES R.", "IRON MAN/TONY STARK"  
"IRON MAN/TONY STARK", "FORTUNE", "DOMINIC"  
"IRON MAN/TONY STARK", "ERWIN", "CLYTEMNESTRA"  
"ERWIN", "CLYTEMNESTRA", "FORTUNE", "DOMINIC"  
"PRINCESS ZANDA", "BLACK PANTHER/T'CHAL"  
"PRINCESS ZANDA", "LITTLE", "ABNER"  
"LITTLE", "ABNER", "BLACK PANTHER/T'CHAL"
```

Apache HBase/
Oracle NoSQL Database

Files/HDFS



Load Graph into Memory from Database

```
pgx> G = session.readGraphWithProperties("graphs/connections_rdbms.json");  
  
{  
  "format": "pg",  
  "db_engine": "RDBMS",  
  "jdbc_url": "jdbc:oracle:thin:@127.0.0.1:1521/mydb",  
  "username": "dbuser",  
  ..  
  "max_num_connections": 8,  
  "name": "connections",  
  "vertex_props": [  
    { "name": "name", "type": "string"},  
    { "name": "role", "type": "string"},  
    { "name": "occupation", "type": "string"},  
    { "name": "country", "type": "string"},  
    { "name": "political", "type": "string"},  
    { "name": "religion", "type": "string"},  
    { "name": "type", "type": "string"}  
  ],  
  "edge_props": [  
    { "name": "weight", "type": "double", "default": "1"}  
  ],  
  "loading":{  
    "load_edge_label": true,  
    "load_vertex_labels": true,  
    "use_vertex_property_value_as_label": "type",  
    "property_value_delimiter": ","  
  }  
}
```

Load from
Property Graph
Schema in
Oracle Database

Load Graph into Memory from NoSQL

```
pgx> G = session.readGraphWithProperties("graphs/connections_nosql.json");  
  
{  
  "format": "pg",  
  "db_engine": "nosql",  
  "hosts": ["localhost:5000"],  
  "store_name": "kvstore",  
  "name": "connections",  
  "vertex_props": [  
    {"name": "name", "type": "string"},  
    {"name": "role", "type": "string"},  
    {"name": "occupation", "type": "string"},  
    {"name": "country", "type": "string"},  
    {"name": "political", "type": "string"},  
    {"name": "religion", "type": "string"},  
    {"name": "type", "type": "string"}  
  "edge_props": [  
    {"name": "weight", "type": "double", "default": "1"}  
  "loading":{  
    "load_edge_label":true,  
    "load_vertex_labels": true,  
    "use_vertex_property_value_as_label": "type",  
    "property_value_delimiter": ","  
}
```

Load Graph into Memory from Files

```
pgx> G = session.readGraphWithProperties("graphs/connections_files.json");  
  
{  
  "format": "flat_file",  
  "separator": ",",  
  "edge_uri_list": ["../../data/connections.ope"],  
  "vertex_uri_list": ["../../data/connections.opv"],  
  "vertex_props": [  
    {"name": "name", "type": "string"},  
    {"name": "role", "type": "string"},  
    {"name": "occupation", "type": "string"},  
    {"name": "country", "type": "string"},  
    {"name": "political", "type": "string"},  
    {"name": "religion", "type": "string"},  
    {"name": "age", "type": "int", "default": "-1"}  
  ],  
  "edge_props": [  
    {"name": "weight", "type": "double", "default": "1000000"}  
  ],  
  "loading": {  
    "load_edge_label": true  
  }  
}
```

Query Graphs

PGQL Graph Query Language

- **Graph pattern matching**

(person) –[:collaborates] -> (person)

- **Basic patterns and reachability patterns**

Can we reach from A to B with an arbitrary number of hops?

- **Familiarity for SQL users**

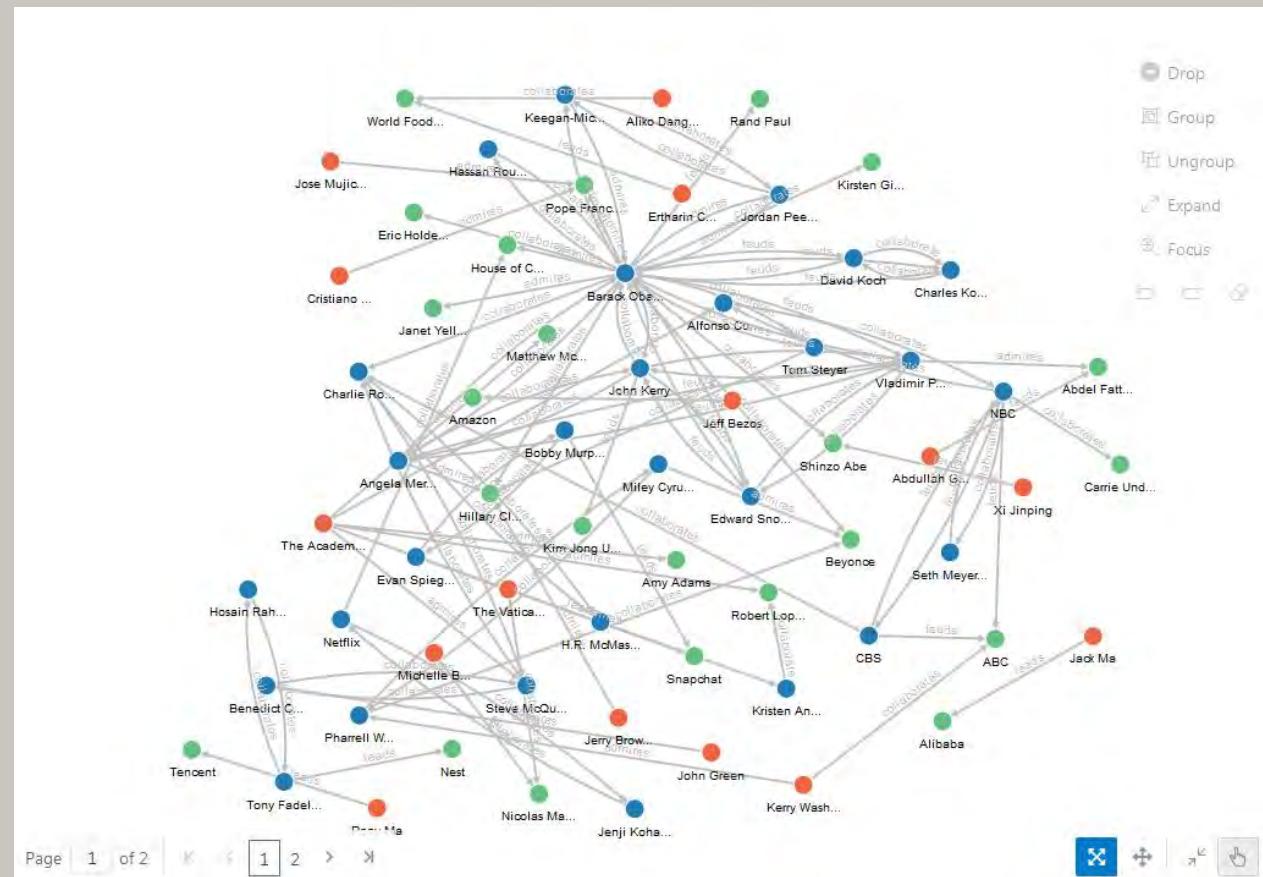
Similar language constructs and syntax

SELECT ... WHEREGROUP BY ... ORDER BY

“Result set” (table) as output

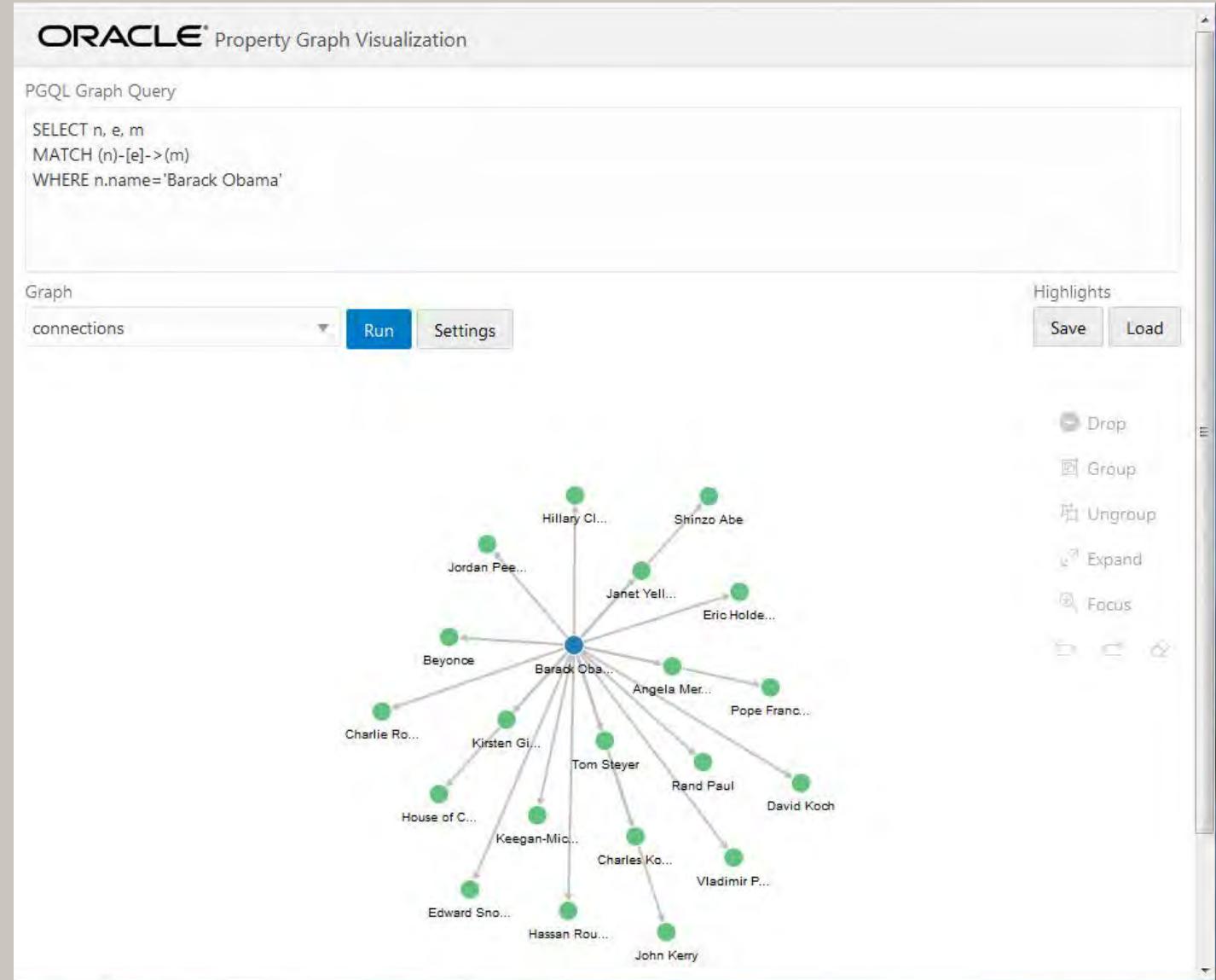
PGOL Examples

```
SELECT e  
MATCH ( )-[e]->( )
```



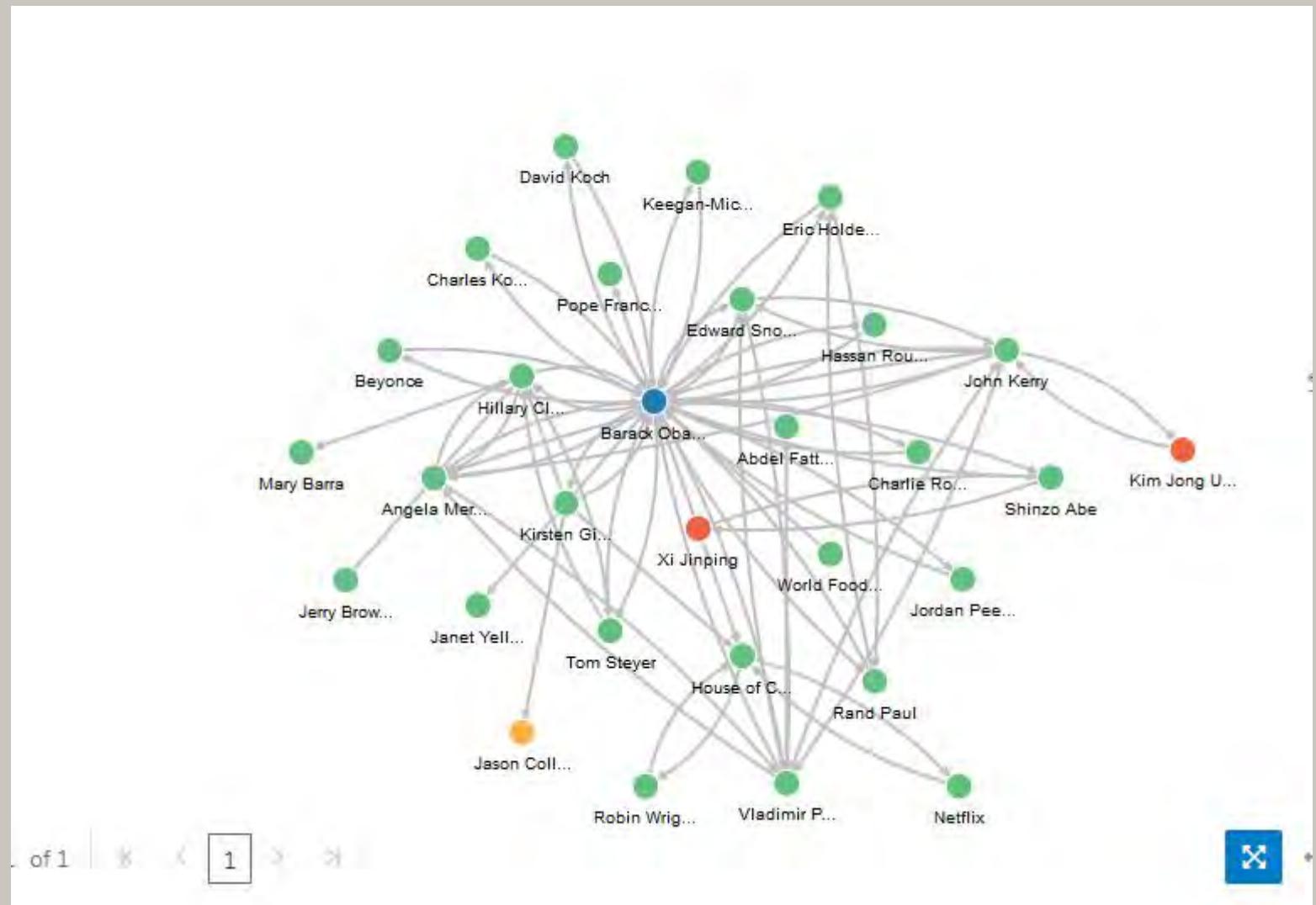
PGQL

```
SELECT e  
MATCH (n)-[e]->(m)  
WHERE n.name='Barack Obama'
```



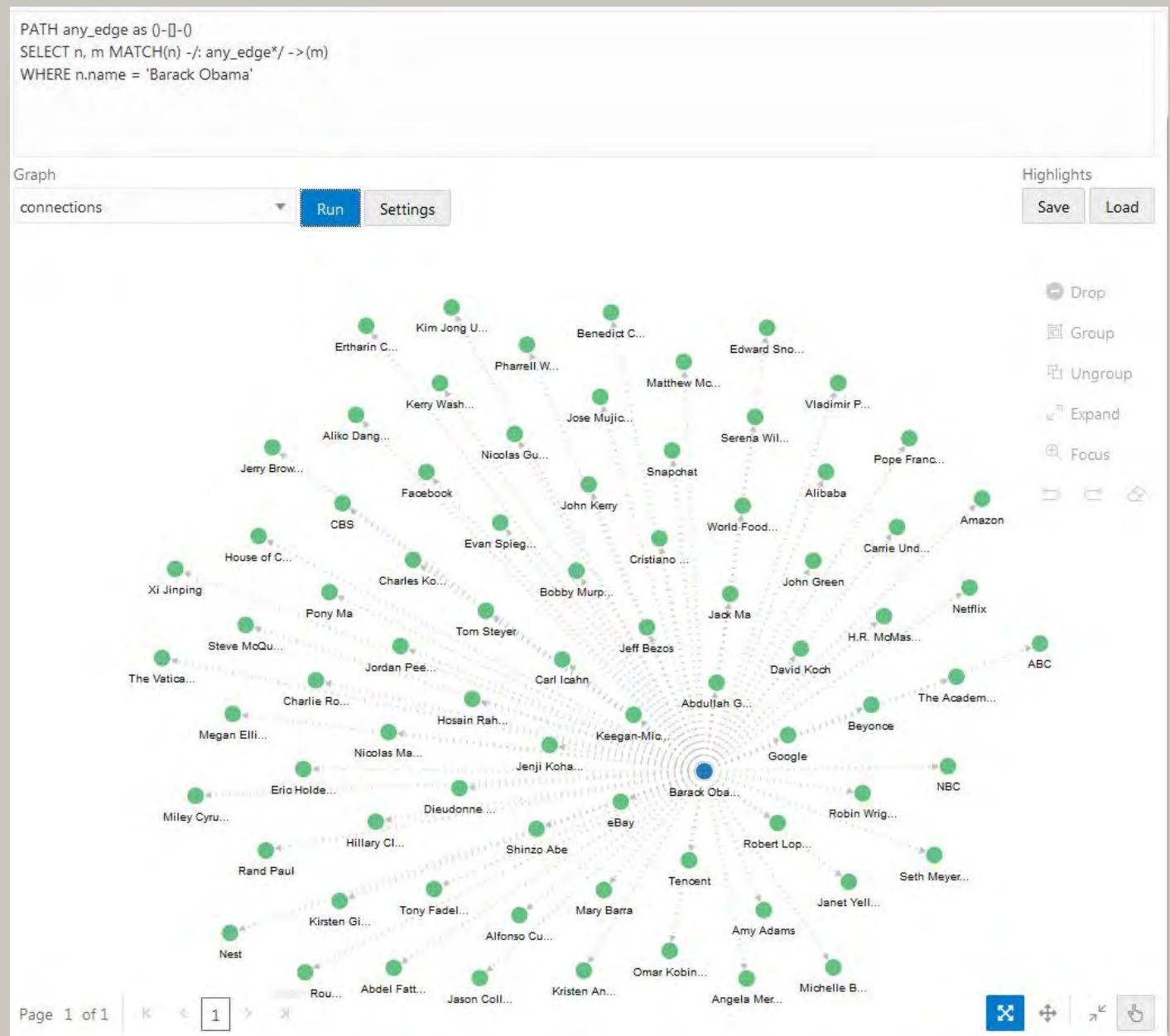
PGQL

Expand connections in
Viz tool



PGQL

```
PATH any_edge as ()-[ ]-()
SELECT n, m MATCH(n) -/: any_edge*/ ->(m)
WHERE n.name = 'Barack Obama'
```



PGQL

```
SELECT label(e), count(*)  
MATCH () -[e]-> ()  
GROUP BY label(e)
```

label(e)	count(*)
admires	28
leads	9
feuds	45
collaborates	82

PGQL

ORACLE® Property Graph Visualization

PGQL Graph Query

```
SELECT n, e, m  
MATCH (n)-[e:collaborates]->(m)
```

Graph

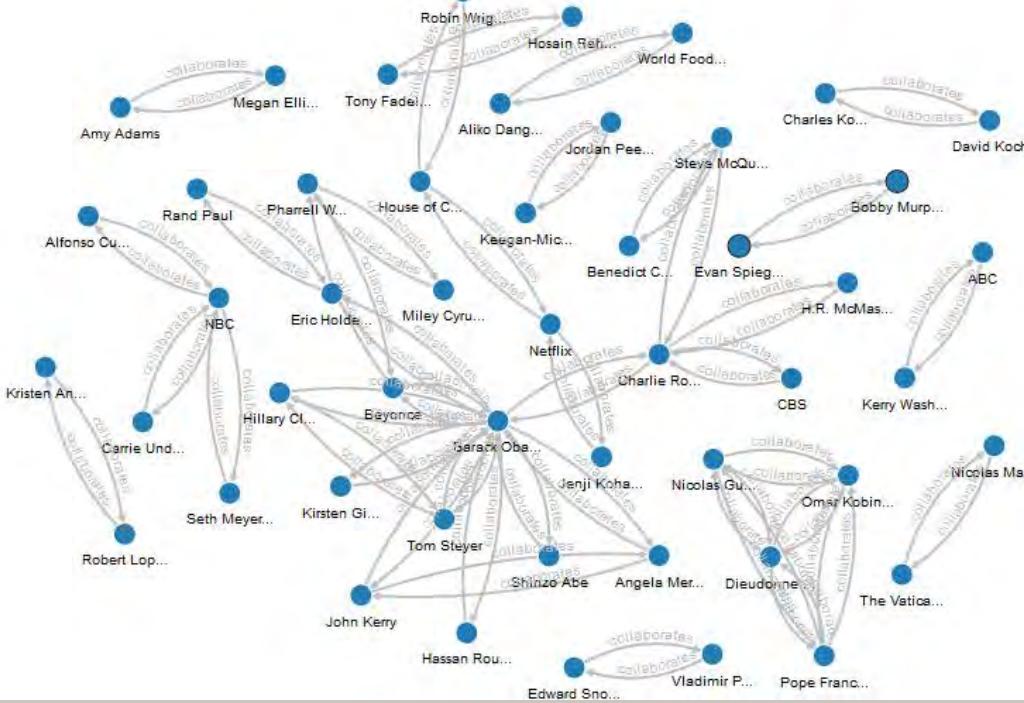
connections

Run

Settings

Highlights

Save Load



Drop

Group

Ungroup

Expand

Focus

...

```
SELECT n, e, m  
MATCH (n)-[e:collaborates]->(m)
```

PGQL

ORACLE® Property Graph Visualization

PGQL Graph Query

```
SELECT n, e, m
MATCH (n)-[e:leads]->(m)
```

Graph

connections

Run

Settings

Highlights

Save

Load

Drop

Group

Ungroup

Expand

Focus

```
graph TD; PopeFrancis((Pope Francis)) -- "1920s" --> EstherinC((Estherin C...)); EstherinC -- "1930s" --> TonyFadel((Tony Fadel...)); TonyFadel -- "1940s" --> TheVatican((The Vatica...)); TheVatican -- "1950s" --> WorldFood((World Food...)); WorldFood -- "1960s" --> Nest((Nest)); Nest -- "1970s" --> Google((Google)); Google -- "1980s" --> JeffBezos((Jeff Bezos)); JeffBezos -- "1990s" --> EvanSpiegel((Evan Spiegel...)); EvanSpiegel -- "2000s" --> BobbyMurphy((Bobby Murp...)); BobbyMurphy -- "2010s" --> Snapchat((Snapchat)); Tencent((Tencent)) -- "1930s" --> Alibaba((Alibaba)); Alibaba -- "1940s" --> Amazon((Amazon)); Amazon -- "1950s" --> PonyMa((Pony Ma)); PonyMa -- "1960s" --> Nest;
```

PGQL

```
SELECT n, e, m, e1, o
MATCH (o) <-[e1:collaborates]-(n)-[e:leads]->(m)
```

ORACLE® Property Graph Visualization

PGQL Graph Query

```
SELECT n, e, m, e1, o
MATCH (o) <-[e1:collaborates]-(n)-[e:leads]->(m)
```

Graph

connections Run Settings

Highlights

Save Load

Drop Group Ungroup Expand Focus

The visualization displays a network graph with various nodes and their connections. The nodes are represented by colored circles (blue, green, red, orange) and are labeled with names and abbreviations. The connections are shown as lines (edges) between the nodes, indicating the relationships defined in the PGQL query.

PGQL

```
SELECT n, e, m  
MATCH (n)-[e:feuds]->(m)
```

ORACLE® Property Graph Visualization

PGQL Graph Query

```
SELECT n, e, m
MATCH (n) - [e:feuds]->(m)
```

Graph

connections Run Settings

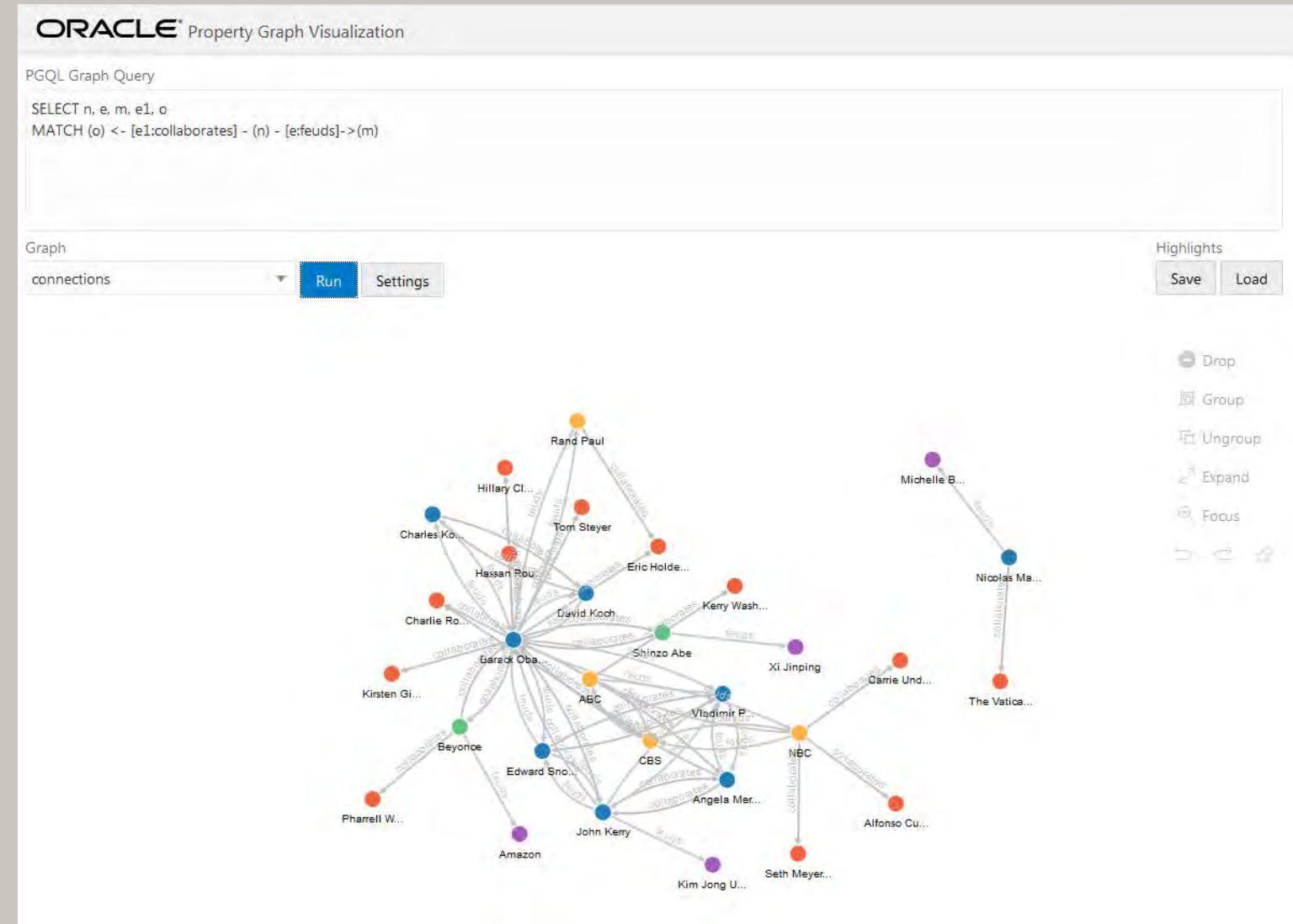
Highlights

Drop Group Ungroup Expand Focus

The figure displays a complex network graph with numerous nodes and edges. Nodes are represented by blue circles, and edges by lines with labels. The network includes a mix of political figures, media outlets, and tech companies. Key nodes include Xi Jinping, Shinzo Abe, Michelle B..., David Koch, Charles Koch, Barack Obama, Rand Paul, Angela Mer..., Edward Snowden, Kim Jong Un, Vladimir P..., John Kerry, NBC, CBS, ABC, Google, Facebook, Alibaba, Amazon, eBay, Tencent, and Snapchat. Edges are labeled with 'feuds' or 'buds', indicating the nature of the relationship between the nodes. The graph is highly interconnected, with many nodes having multiple connections to others.

PGQL

```
SELECT n, e, m, e1, o
MATCH (o) <-[e1:collaborates]-(n)-[e:feuds]->(m)
```



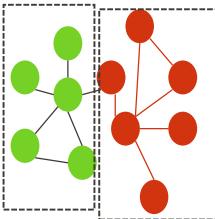
Analyze Graphs

Analyze Graphs

- What are important nodes?
- What is the shortest path between these two nodes?
- What are disconnected clusters in a graph?
-

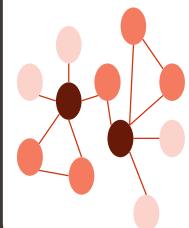
Graph Analytics Algorithms

Detecting Components and Communities



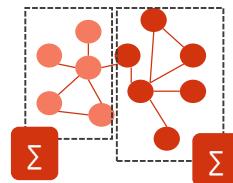
Tarjan's, Kosaraju's, Weakly Connected Components, Label Propagation (w/ variants), Soman and Narang's Specification

Ranking and Walking



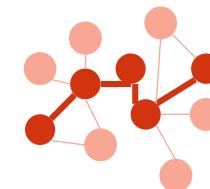
Pagerank, Personalized Pagerank, Betweenness Centrality (w/ variants), Closeness Centrality, Degree Centrality, Eigenvector Centrality, HITS, Random walking and sampling (w/ variants)

Evaluating Community Structures



Conductance, Modularity, Clustering Coefficient (Triangle Counting), Adamic-Adar

Path-Finding



Hop-Distance (BFS), Dijkstra's, Bi-directional Dijkstra's, Bellman-Ford's

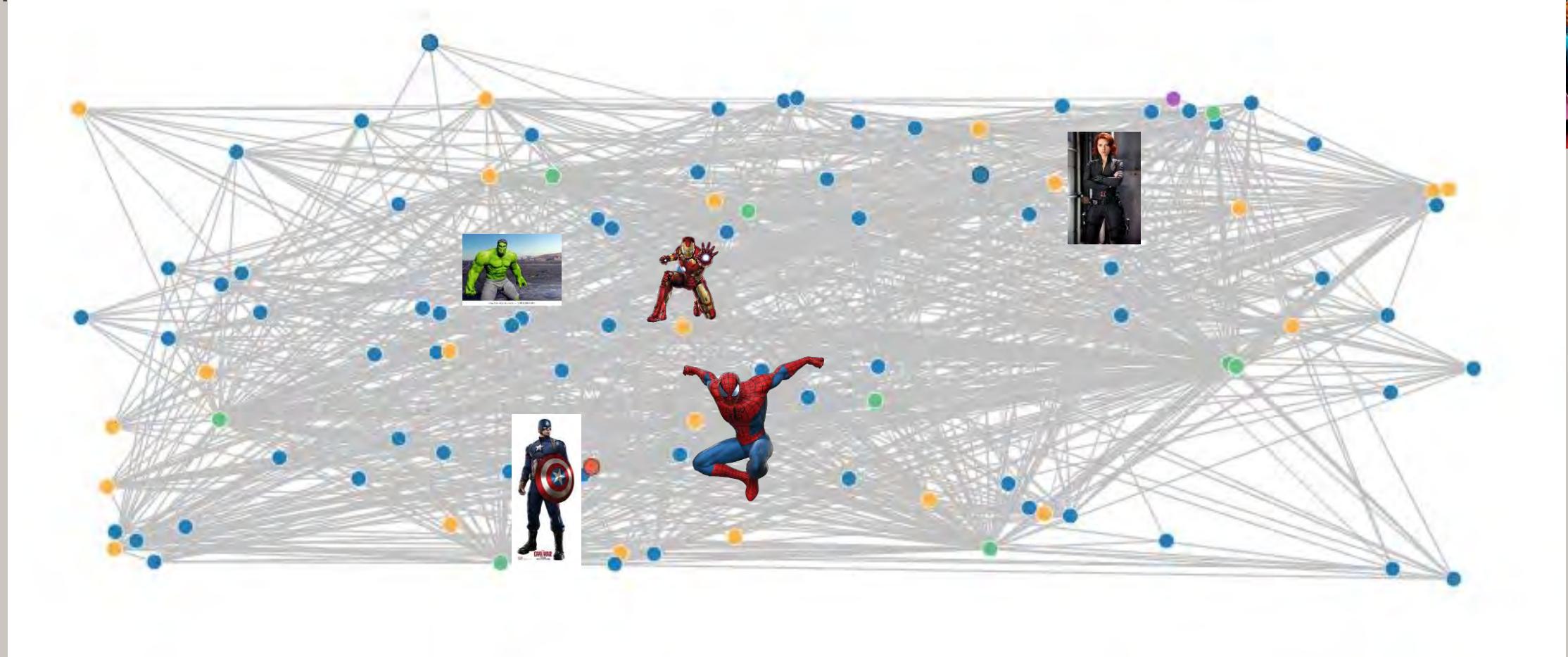
Link Prediction

SALSA
(Twitter's Who-to-follow)

Other Classics

Vertex Cover, Minimum Spanning-Tree (Prim's)

Marvel Superheros Dataset



Ana

PGQL Graph Query

```
1 SELECT n, n0, n1, e0, e1, e2, n.pageRank, n0.pageRank, n1.pageRank
2 MATCH (n)-[e0]-(n0)-[e1]-(n1), (n)-[e2]-(n1)
3 WHERE ID(n0) = 'IRON MAN/TONY STARK'
4 ORDER BY n.pageRank DESC, n0.pageRank DESC, n1.pageRank DESC LIMIT 30
```

Graph

sub-graph_5

Run

Settings

Highlights

Save

Load

Drop

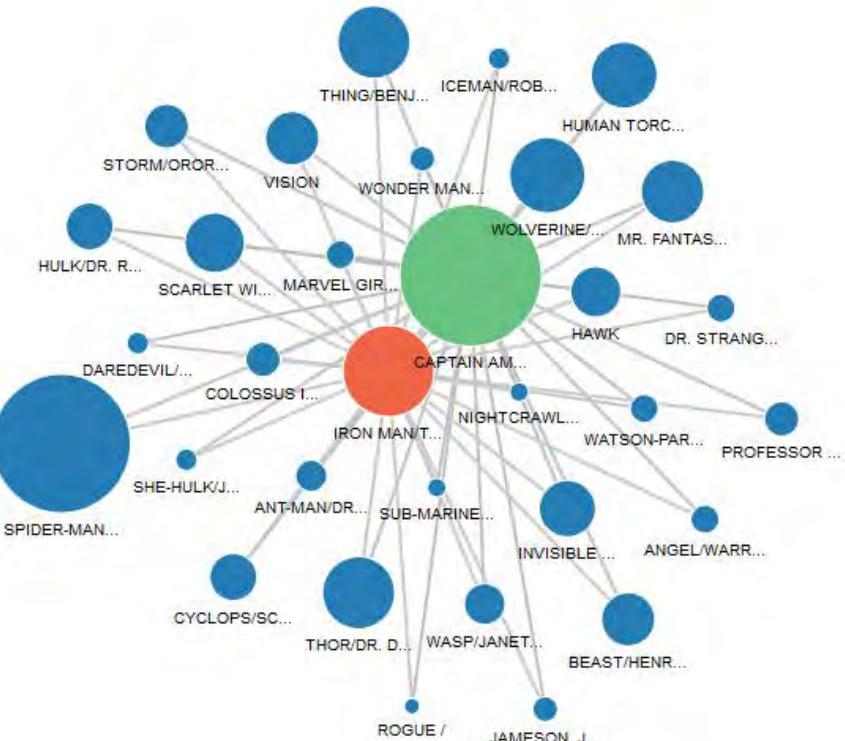
Group

Ungroup

Expand

Focus

Zoom



```
g = session
analyst.pa
analyst ve
g.publish()
```

Graph Attributes

Vertices

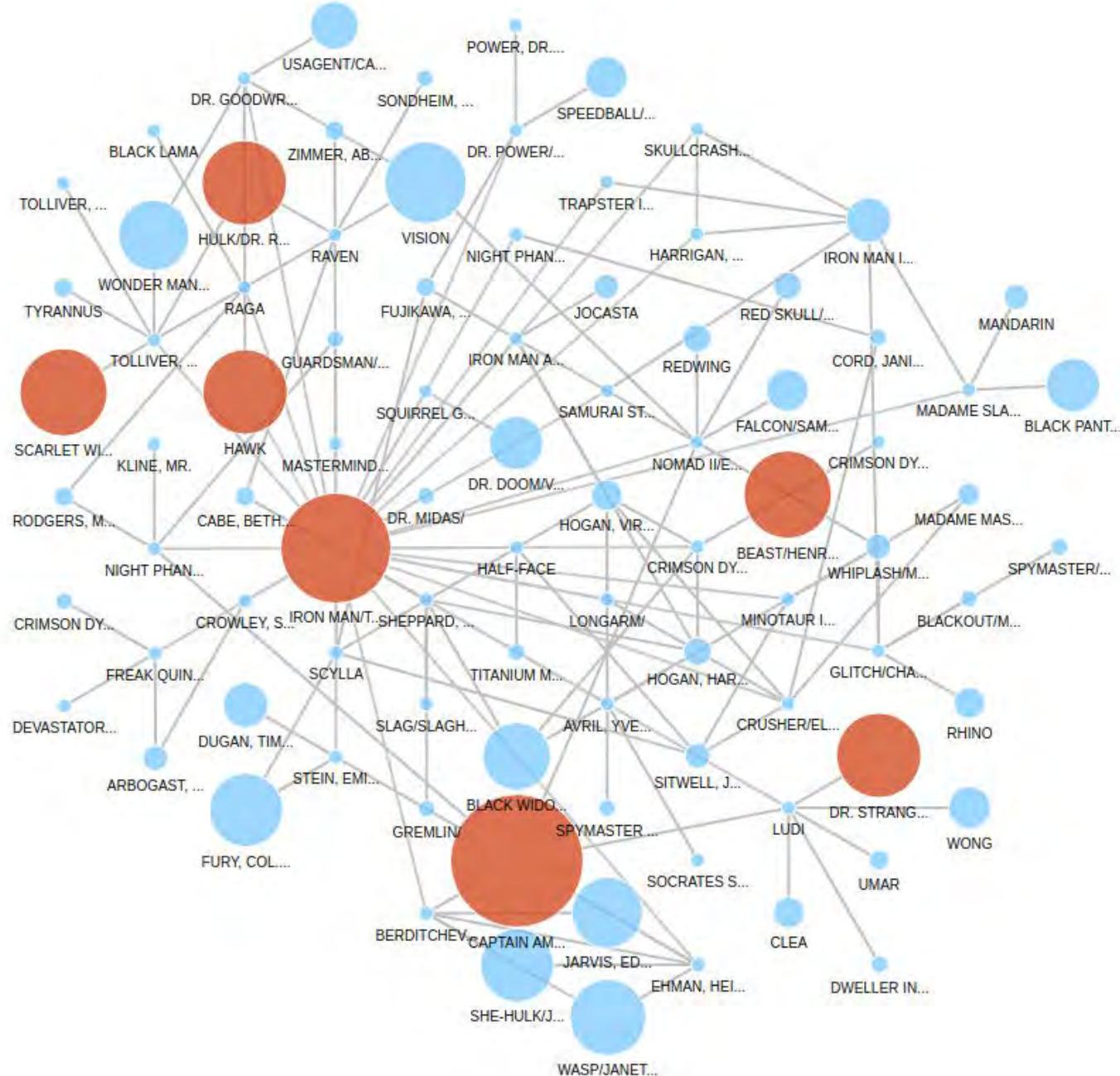
- n1
- n
- n0

Edges

Page 1 of 1

1





PGQL Graph Query

```

1 SELECT n, n0, n1, e0, e1, n.betweenness, n0.betweenness, n1.betweenness
2 MATCH (n)-[e0]-(n0)-[e1]-(n1)
3 WHERE ID(n) = 'SPIDER-MAN/PETER PAR' AND n1.betweenness > 0 AND n0.betweenness > 0
4 ORDER BY n.betweenness DESC, n0.betweenness ASC, n1.betweenness ASC LIMIT 500

```

Graph

sub-graph_5

Run

Settings



Highlights

Save

Load

Drop

Group

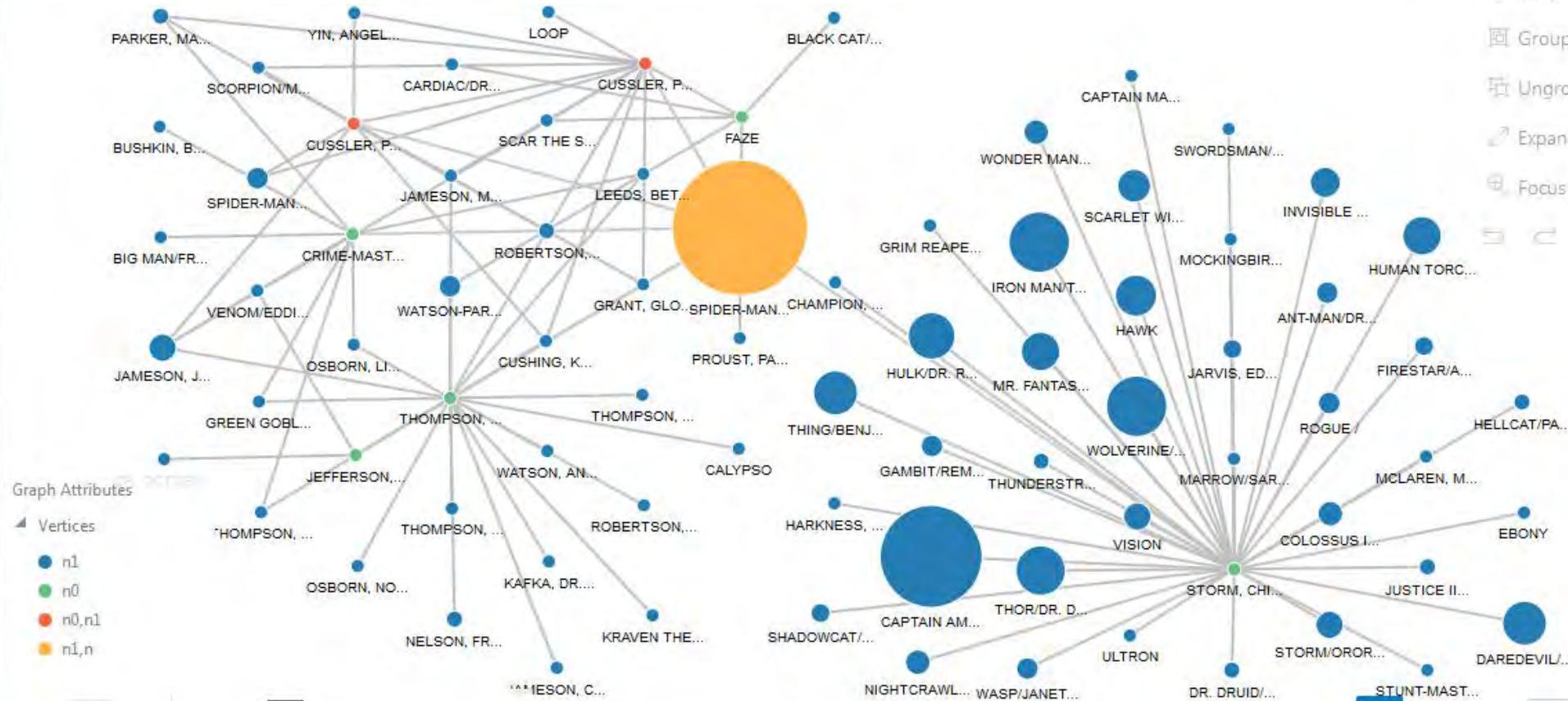
Ungroup

Expand

Focus

Edit

Edit



Page 1 of 5

1 2 3 4 5 > <



Com

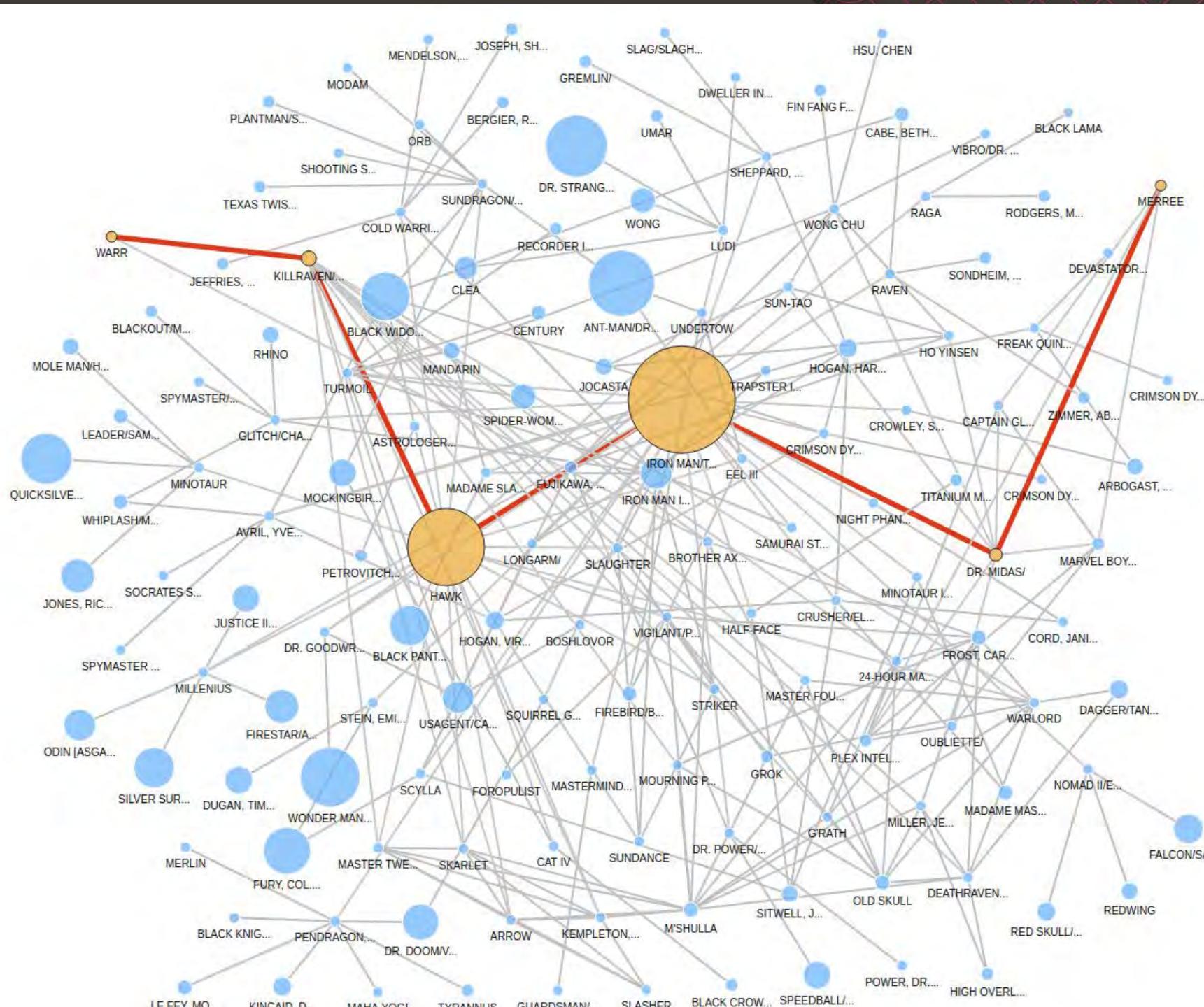
Find sh

```
w = g.createEdge()
w.fill((Double) 1.0)
src = g.getVertex(0)
dst = g.getVertex(1)
path = analyst.getPath(src, dst)

// Creating both paths
in_path = g.createPath()
for (PgxEEdge e : edges)
    in_path.setEdge(e)
}

v_in_path = g.createPath()
for (PgxEVertex v : vertices)
    v_in_path.setVertex(v)
}

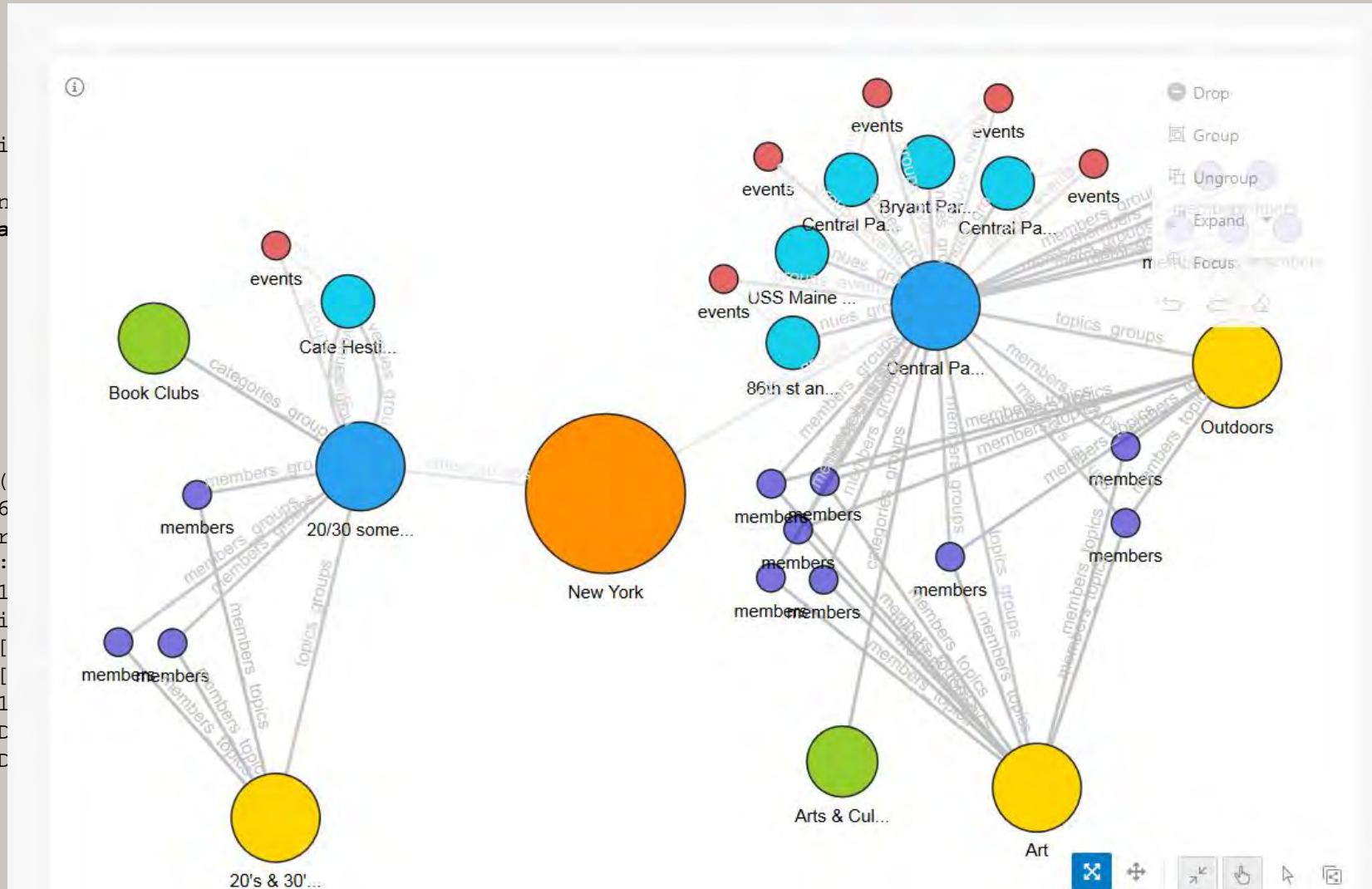
// Shortest Path
SELECT n, e, m,
n.v_in_path, m.v_in_path
MATCH (n)-[e]-(m)
WHERE e.in_path = true
ORDER BY n.page
```



Meetup Dataset

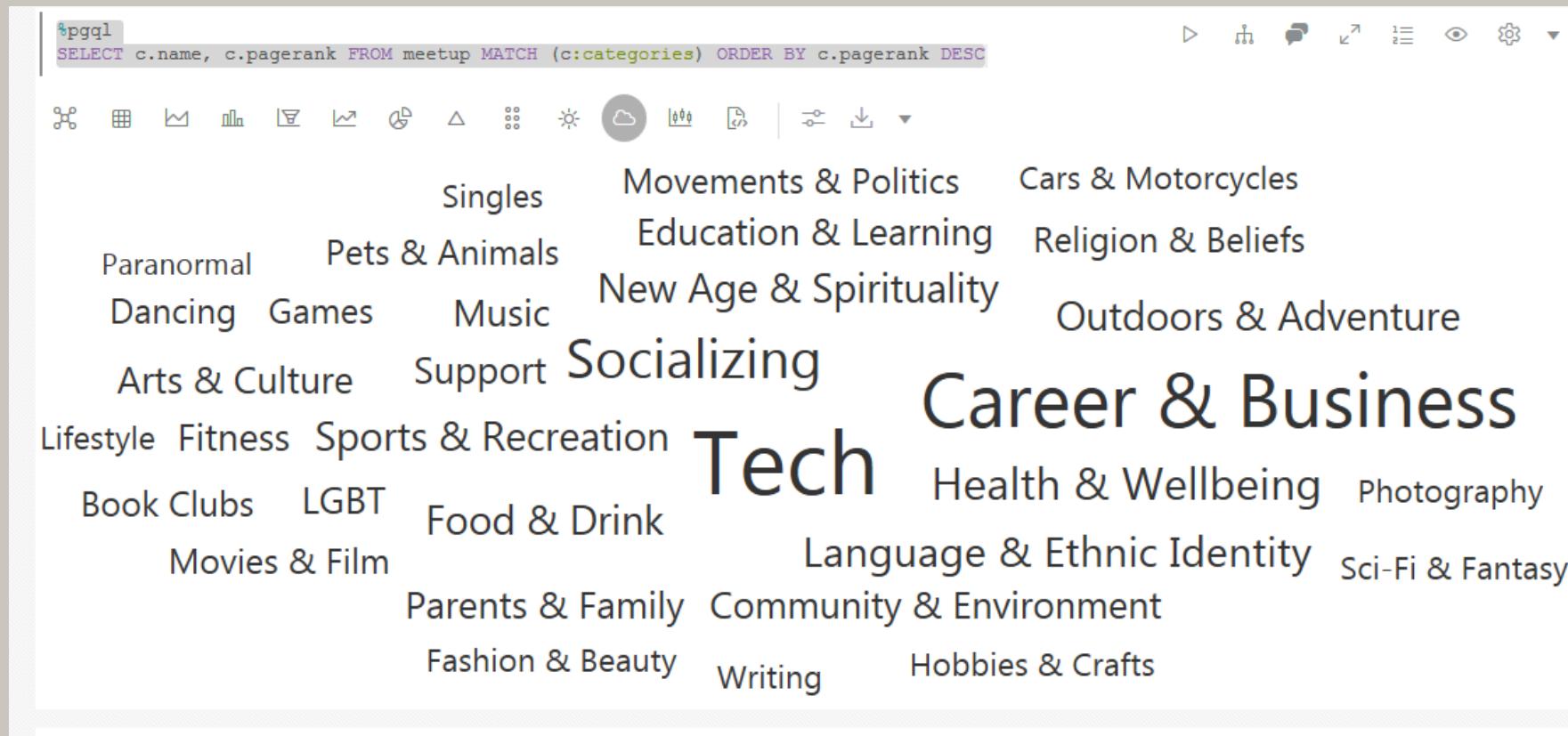
```
%pgx-java
PgxBatchSession sessi
var new_analyst =
VertexProperty<Lon
new_analyst.pageRa
```

```
%pgql
SELECT venue, gr,
FROM meetup MATCH(
(event:events)-[e6
(gr:groups)-[e7:gr
(city:cities)-[e3:
(topic:topics)-[e1
(category:category
(member:members)-[
(member:members)-[
WHERE (gr.id = 641
(gr.id = 10504 AND
(gr.id = 15352 AND
gr.id = 8199
```



Meetup Dataset

```
%pgql
SELECT c.name, c.pageRank FROM meetup MATCH (c:categories) ORDER BY c.pageRank DESC
```





Clothing and Fashion Designers
Women Business Networking
Social Marketing Real Estate Investment Education Corporate Social Responsibility
Trader Education Technology Women's Business Networking Investment Education
B2B Networking Social Higher Education Financial Education
Social Enterprise Social Media Other Small Business Networking Social Networking Financial Technology
Social Entrepreneurship Social Media Marketing Entertainment Industry
Arts & Entertainment Business Networking International (BNI)
Technology Consulting Social, Social, Social

Business Referral Networking

Professional Networking

Query and Analyze Using Notebook

Getting Started with Graph A...



1. Connect to the Graph

FINISHED

Took 0 sec. Last updated by anonymous at April 12 2019, 6:31:12 PM.

Access the published graph

FINISHED

```
%pgx
pg = session.getGraph("connections");

PgxGraph[name=connections,N=78,E=164,created=1567704926862]
```

Took 11 sec. Last updated by anonymous at September 05 2019, 1:50:20 PM.

2. Explore the Graph

FINISHED

Took 0 sec. Last updated by anonymous at April 12 2019, 6:31:12 PM.

How many vertices ?

FINISHED

```
%pgx
pg.getNumVertices()
```

78

Took 0 sec. Last updated by anonymous at September 05 2019, 1:50:27 PM.

How many edges ?

FINISHED

```
%pgx
pg.getNumEdges()
```

164

Took 0 sec. Last updated by anonymous at September 05 2019, 1:50:29 PM.

Getting Started with Graph Analyti...



3. Use Built-in Analytics

FINISHED ▶

Took 0 sec. Last updated by anonymous at September 05 2019, 1:56:00 PM.

Page Rank

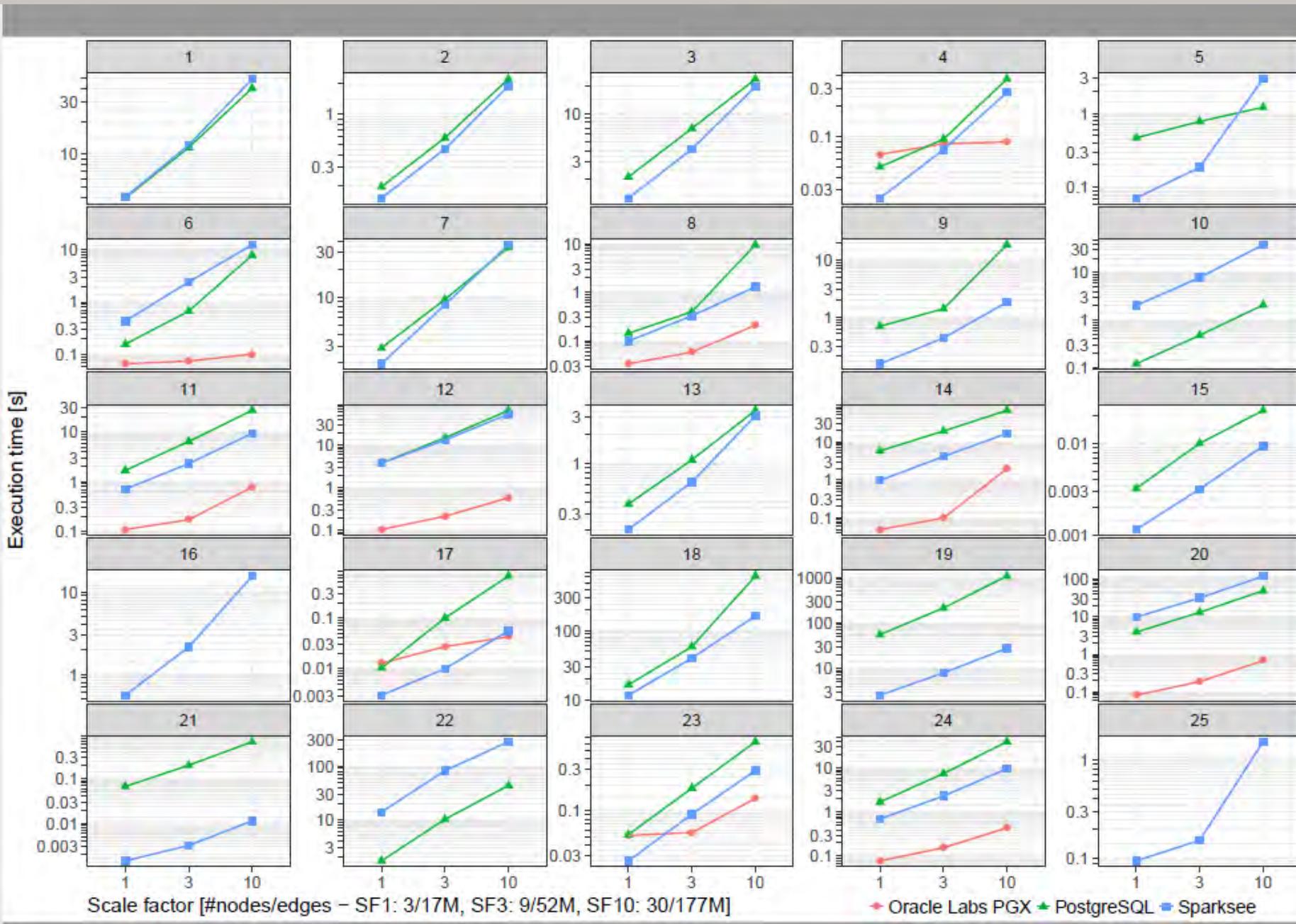
FINISHED ▶

```
%pgx
r=analyst.pageRank(graph:pg, max:1000, variant:'APPROXIMATE')
r.getTopKValues(10)
```



Took 0 sec. Last updated by anonymous at September 05 2019, 1:56:27 PM: (outdated)

Performance and Scale



LDBC benchmark

Scale factor 1

Number of vertices: 3,181,724
 Number of edges: 17,256,038
Memory footprint: 1.3GB

Scale factor 3

Number of vertices: 9,281,922
 Number of edges: 52,695,735
Memory footprint: 4.0GB

Scale factor 10

Number of vertices: 29,987,835
 Number of edges: 176,623,445
Memory footprint: 13.3 GB

Oracle

Postgres SQL

Sparksee



Property Graph Sizing Recommendations

Table 1-1 Property Graph Sizing Recommendations

Graph Size	Recommended Physical Memory to be Dedicated	Recommended Number of CPU Processors
10 to 100M edges	Up to 14 GB RAM	2 to 4 processors, and up to 16 processors for more compute-intensive workloads
100M to 1B edges	14 GB to 100 GB RAM	4 to 12 processors, and up to 16 to 32 processors for more compute-intensive workloads
Over 1B edges	Over 100 GB RAM	12 to 32 processors, or more for especially compute-intensive workloads

Graph Cloud Service

Graph Cloud Service

Fully managed

“One-click” deployment: no installation, zero configuration

Automated failure detection and recovery

Automated graph modeler

Easily convert your relational data into property graphs

Pre-built algorithms, flows and interactive queries

Java

PGQL

Rest APIs

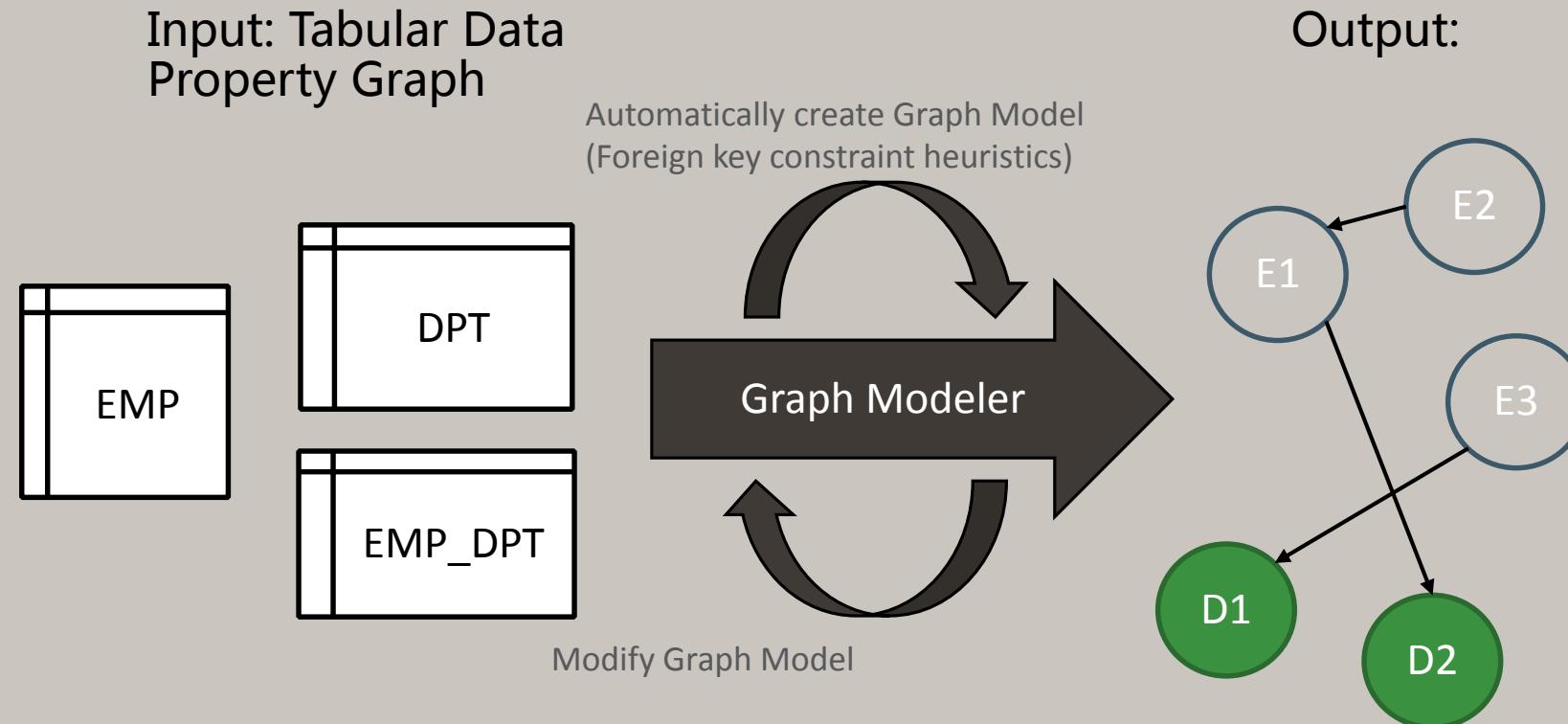
Rich User Interface

Low code / zero code features

Notebook support and powerful data visualization features



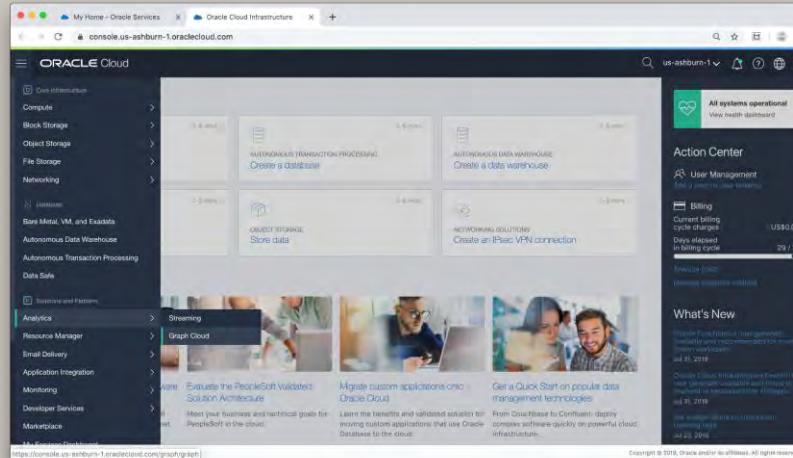
Automated graph modeling



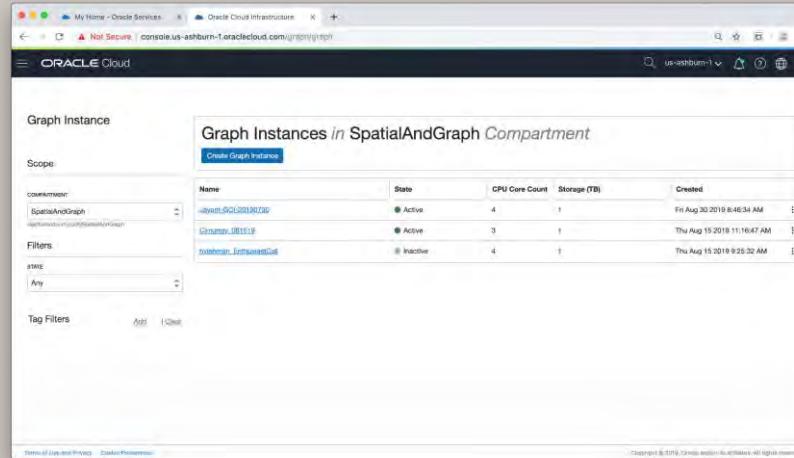
Sample graph cloud workflow

Provision instance
Model and load graph
Analyze, visualize, share results

Provision an instance



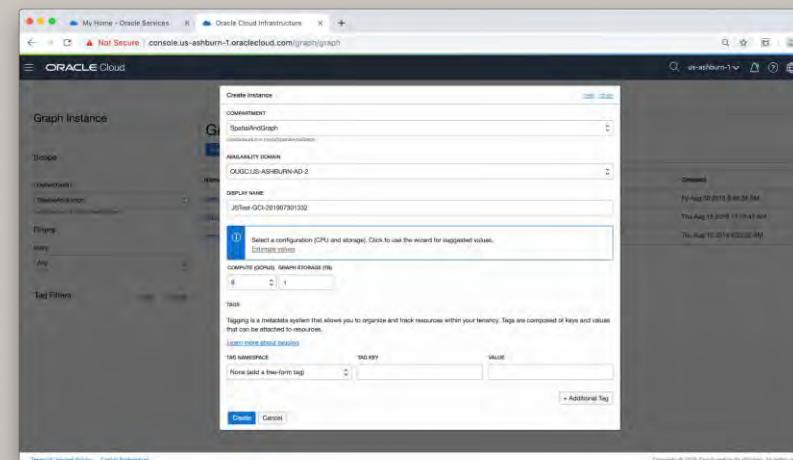
The screenshot shows the Oracle Cloud Infrastructure home page. The left sidebar lists various services: Compute, Block Storage, Object Storage, File Storage, Networking, Bare Metal, VM, and Ephemeral, Autonomous Data Warehouse, Autonomous Transaction Processing, Data Safe, Analytics, Resource Manager, Email Delivery, Application Integration, Monitoring, Developer Services, and Marketplace. The 'Analytics' section is currently selected, showing sub-options for Streaming and Graph Cloud. The main content area displays several cards: 'AUTONOMOUS TRANSACTION PROCESSING' (Create a database), 'AUTONOMOUS DATA WAREHOUSE' (Create a data warehouse), 'Bare Metal, VM, and Ephemeral' (Store data), and 'Networking' (Create an IPsec VPN connection). A 'What's New' section highlights Oracle Database 19c and Oracle Database 18c. The bottom of the page includes a footer with links to Terms of Use and Privacy and Oracle Preferences.



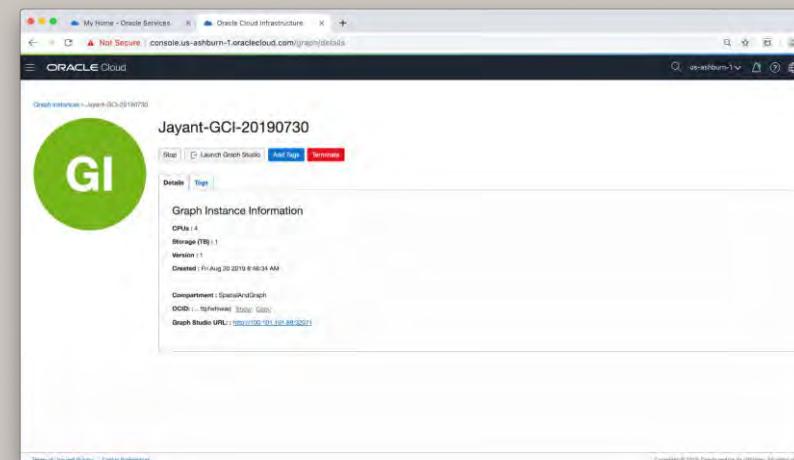
The screenshot shows the 'Graph Instances in SpatialAndGraph Compartment' page. The top navigation bar indicates the compartment is 'SpatialAndGraph'. The main content area displays a table of graph instances:

Name	State	CPU Core Count	Storage (TB)	Created
jayant-GCI-20190730	Active	4	1	Fri Aug 30 2019 8:46:34 AM
Crnay-GCI-20190815	Active	3	1	Thu Aug 15 2019 11:16:47 AM
runman_TempuseCall	Inactive	4	1	Thu Aug 15 2019 9:25:32 AM

Below the table are 'Scope' and 'Filters' dropdowns, and a 'Tag Filters' section. The bottom of the page includes a footer with links to Terms of Use and Privacy and Oracle Preferences.



The screenshot shows the 'Create instance' wizard for the 'Graph Cloud' service. The first step, 'COMPARTMENT', shows the compartment is 'SpatialAndGraph'. The second step, 'AVAILABILITY DOMAIN', shows the availability domain is 'OACLOUD-ASHBURN-AD-2'. The third step, 'DISPLAY NAME', shows the display name is '2019-GCI-20190730132'. The fourth step, 'Select a configuration (CPU and storage)', shows the configuration is '4 CPU, 1 TB Storage'. The fifth step, 'Tags', shows a table of tags: 'TAG NAME' (None), 'TAG KEY' (None), and 'TAG VALUE' (None). The bottom of the page includes a footer with links to Terms of Use and Privacy and Oracle Preferences.



The screenshot shows the 'Graph instance > Jayant-GCI-20190730' details page. The top navigation bar shows the compartment is 'SpatialAndGraph'. The main content area displays the following information:

Graph Instance Information

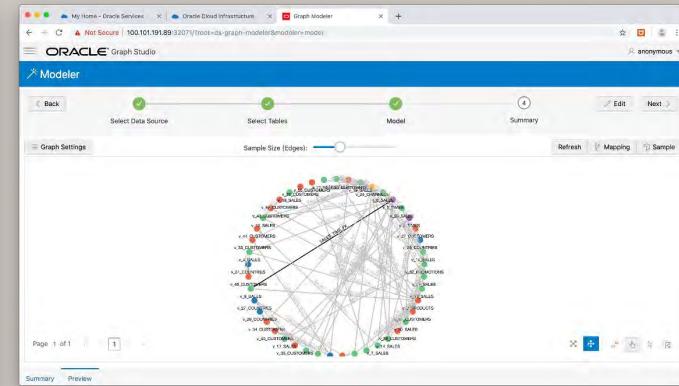
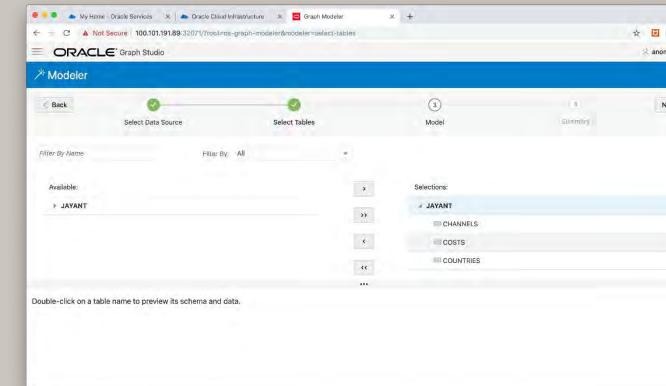
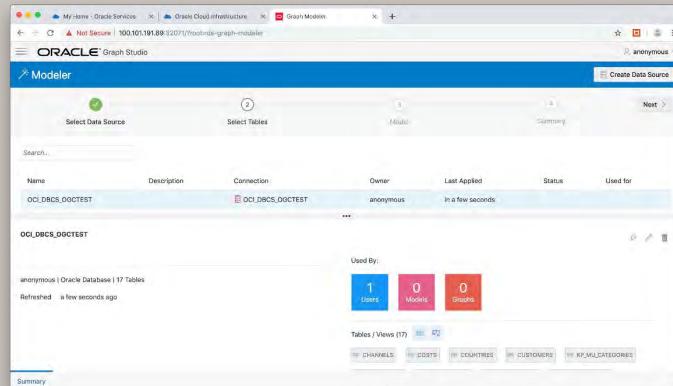
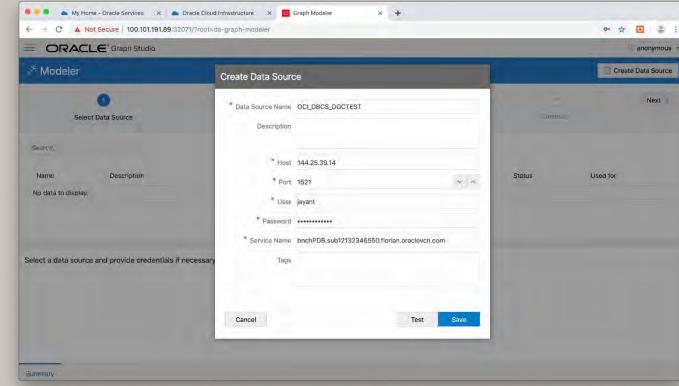
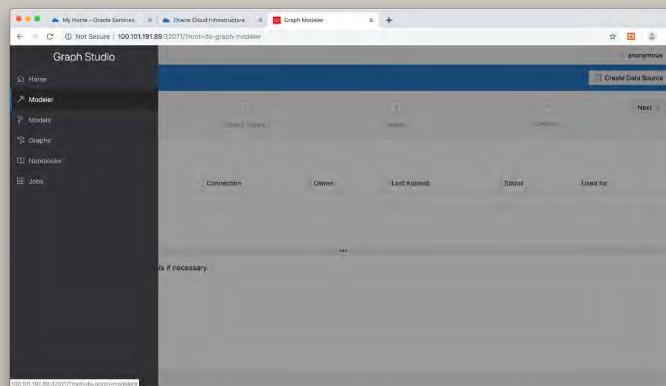
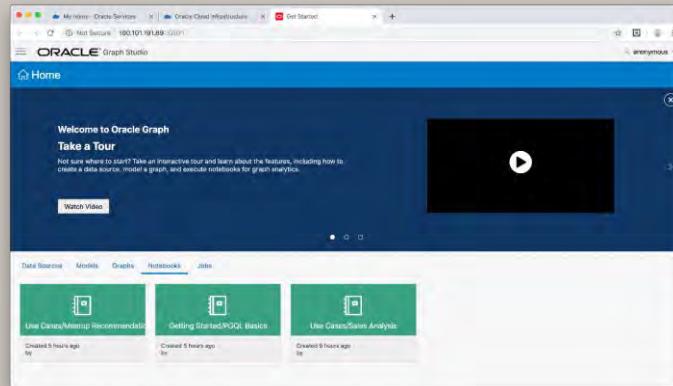
- CPU: 4
- Storage (TB): 1
- Version: 1
- Created: Fri Aug 30 2019 8:46:34 AM

Graph Instance Details

- Compartment: SpatialAndGraph
- OCID: ocid1.graphinstance.oc1.ashburn1.20190730132
- Graph Studio URL: <https://ocid1.graphstudio.oc1.ashburn1.20190730132>

The bottom of the page includes a footer with links to Terms of Use and Privacy and Oracle Preferences.

Connect to data source, create graph



My Home - Oracle Services Oracle Cloud Infrastructure Graph Modeler

Not Secure | 100.101.191.89:32071/?root=ds-graph-modeler

Graph Studio

anonymous

Create Data Source

Home

Modeler

Models

Graphs

Notebooks

Jobs

Connection

Owner

Last Applied

Status

Used for

Next

...
ls if necessary.

100.101.191.89:32071/?root=ds-graph-modeler

My Home - Oracle Services Oracle Cloud Infrastructure Graph Modeler

Not Secure | 100.101.191.89:32071/?root=ds-graph-modeler

ORACLE Graph Studio anonymous

Modeler Create Data Source

Create Data Source

1 Select Data Source

Search:

Name	Description	Status	Used for
No data to display.			

Select a data source and provide credentials if necessary

Create Data Source

* Data Source Name OCI_DBCS_OGCTEST

Description

* Host 144.25.39.14

* Port 1521

* User jayant

* Password *****

* Service Name bnchPDB.sub12132346550.florian.oraclevcn.com

Tags

Cancel Test Save

Summary

My Home - Oracle Services | Oracle Cloud Infrastructure | Graph Modeler

Not Secure | 100.101.191.89:32071/?root=ds-graph-modeler

ORACLE Graph Studio

Modeler

Create Data Source

1 Select Data Source 2 Select Tables 3 Model 4 Summary Next >

Search...

Name	Description	Connection	Owner	Last Applied	Status	Used for
OCI_DBCS_OGCTEST		OCI_DBCS_OGCTEST	anonymous	in a few seconds		

OCI_DBCS_OGCTEST

anonymous | Oracle Database | 17 Tables

Refreshed a few seconds ago

Used By:

1 Users 0 Models 0 Graphs

Tables / Views (17) CHANNELS COSTS COUNTRIES CUSTOMERS KP_MU_CATEGORIES

Summary

My Home - Oracle Services X | Oracle Cloud Infrastructure X | Graph Modeler X +

Not Secure 100.101.191.89:32071/?root=ds-graph-modeler&modeler=select-tables

ORACLE Graph Studio anonymous

Modeler

Back Select Data Source Select Tables Model Summary Next >

Filter By Name Filter By: All

Available: JAYANT

Selections: JAYANT CHANNELS COSTS COUNTRIES

Double-click on a table name to preview its schema and data.

Confidential – Oracle Restricted

My Home - Oracle Services | Oracle Cloud Infrastructure | Graph Modeler

Not Secure | 100.101.191.89:32071/?root=ds-graph-modeler&modeler=model

ORACLE Graph Studio anonymous

Modeler

Back Select Data Source Select Tables Model Summary

Graph Settings Sample Size (Edges): 4000 Refresh Mapping Sample

Page 1 of 1 | 1

Summary Preview

Analyze and visualize

My Home - Oracle Services | Oracle Cloud Infrastructure | Use Cases/Sales Analysis | Not Secure | +

ORACLE Graph Studio

Use Cases > Sales Analysis

What other customers bought the same products through the same channel?

To showcase a little bit more complicated pattern match, here a query which visualizes what other customers bought the same products as the given customer through the same channel.

```
spgql
select a, b, c, d, e, f, other_customer.name
from sales
match (customer:customer) <--[a]--> (sale1:channel:products) <--[c]--> (sale2) -[d]--> (other_customer:customer)
      (sale1) -[e]--> (product1:products), (sale2) -[f]--> (product2:products), (customer)
      on sale1.amount_sold > 1000.0
      and sale2.amount_sold > 1000.0
      and product1.product = product2
      limit 20
customer_id
3221
```

Vertices:

- channel
- sale2
- other_customer
- customer
- sale1

The graph visualization displays a network of nodes representing customers and their purchases. Nodes are color-coded: red for customers, blue for products, and green for channels. Edges represent purchases, with some edges being thicker than others. A legend on the left identifies the node types: channel (green), sale2 (red), other_customer (blue), customer (blue), and sale1 (red). A tooltip on the right indicates a query for '20x30 some...'. The overall layout is a dense cluster of nodes with connecting lines.

My Home - Oracle Services | Oracle Cloud Infrastructure | Use Cases/Meetup Recommendation | Not Secure | +

ORACLE Graph Studio

Use Cases > Meetup Recommendation

So, if we expand some of the rows from the previous tables using the graph schema from above, we generate a graph structure like the following one:

Vertices

- category
- topic
- member
- event
- venue
- gr
- city

Edges

The graph visualization shows a complex network of nodes and edges. Nodes are categorized by color: green for 'category', yellow for 'topic', blue for 'member', red for 'event', cyan for 'venue', orange for 'city', and purple for 'gr'. Edges represent relationships like 'members' and 'members'. A large orange node labeled 'New York' is at the center, connected to various other nodes. A legend on the left identifies the node types. A tooltip on the right indicates a query for '20x30 some...'. The overall structure is a dense web of interconnected nodes.

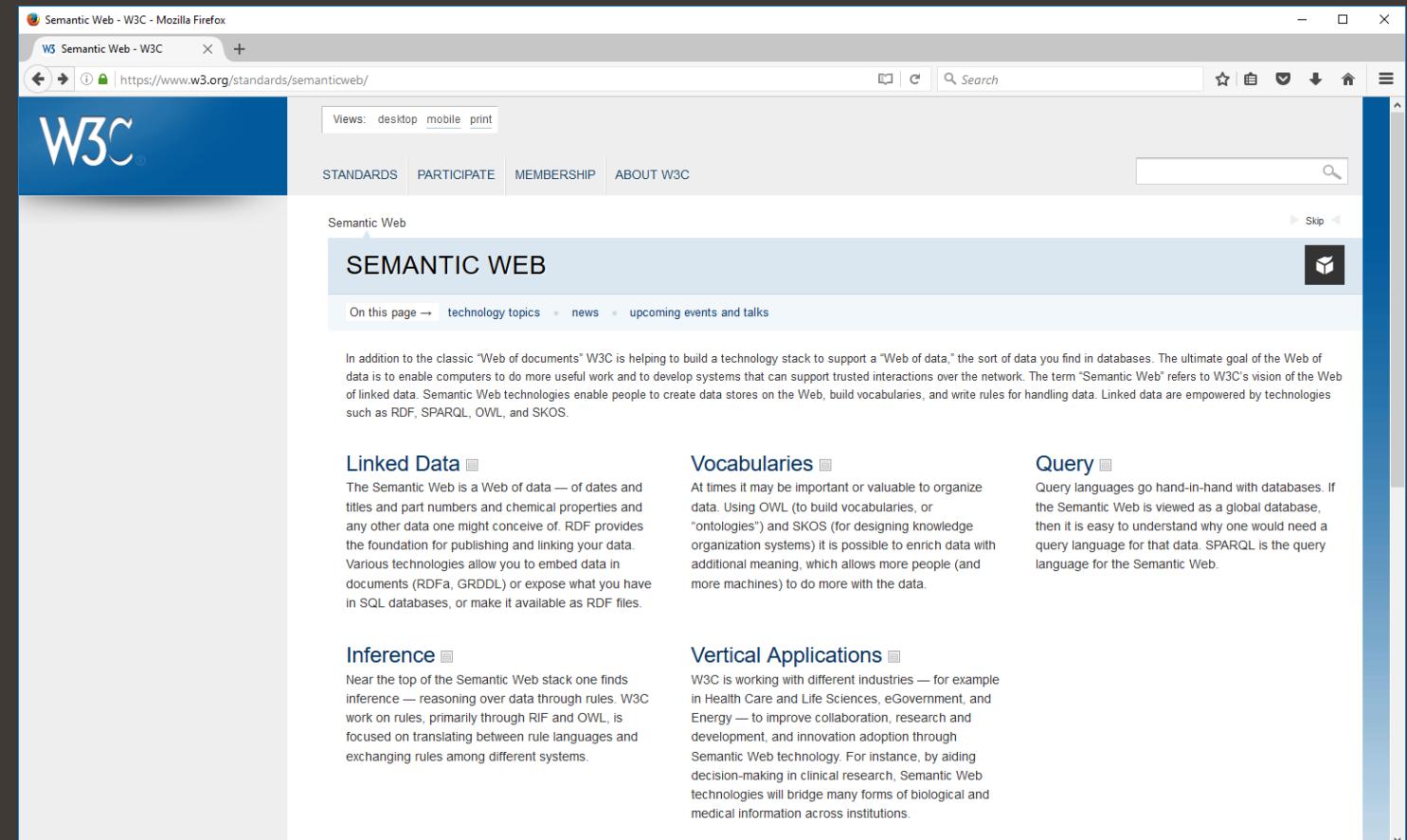
RDF Graphs

W3C Standards for Knowledge Graphs

The World Wide Web Consortium has defined a suite of standards to support Linked Data and Knowledge Graphs.

Fundamental Concepts are:

- URIs
- Links to other resources
- Standard Data Model (RDF)
- Standard Ontology Language (OWL)
- Standard Query (SPARQL)



The screenshot shows a Mozilla Firefox browser window displaying the W3C Semantic Web standards page. The URL in the address bar is <https://www.w3.org/standards/semanticweb/>. The page features a blue header with the W3C logo and navigation links for STANDARDS, PARTICIPATE, MEMBERSHIP, and ABOUT W3C. Below the header, a main content area is titled "SEMANTIC WEB". It includes a sub-section "On this page" with links to "technology topics", "news", and "upcoming events and talks". The main content discusses the goal of the Semantic Web to enable a "Web of data" and lists several key concepts: Linked Data, Vocabularies, Inference, and Vertical Applications, each with a brief description and a "read more" link.

Key Features

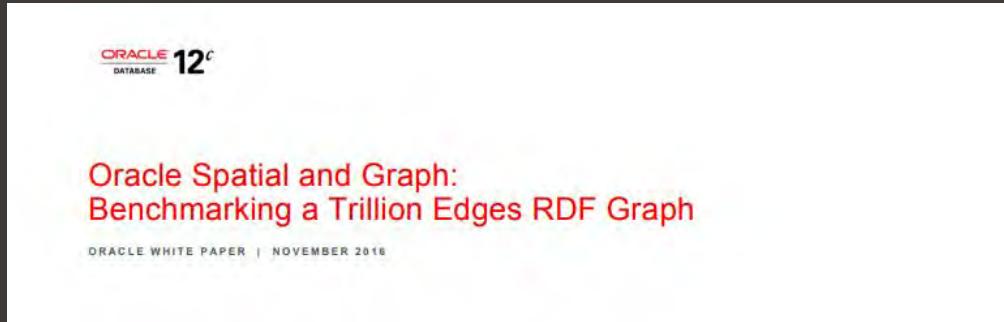
Scalable RDF Graph database

Scales to billions of nodes and edges

Full standards support

RDF, OWL, SPARQL

Integrated with Property Graphs

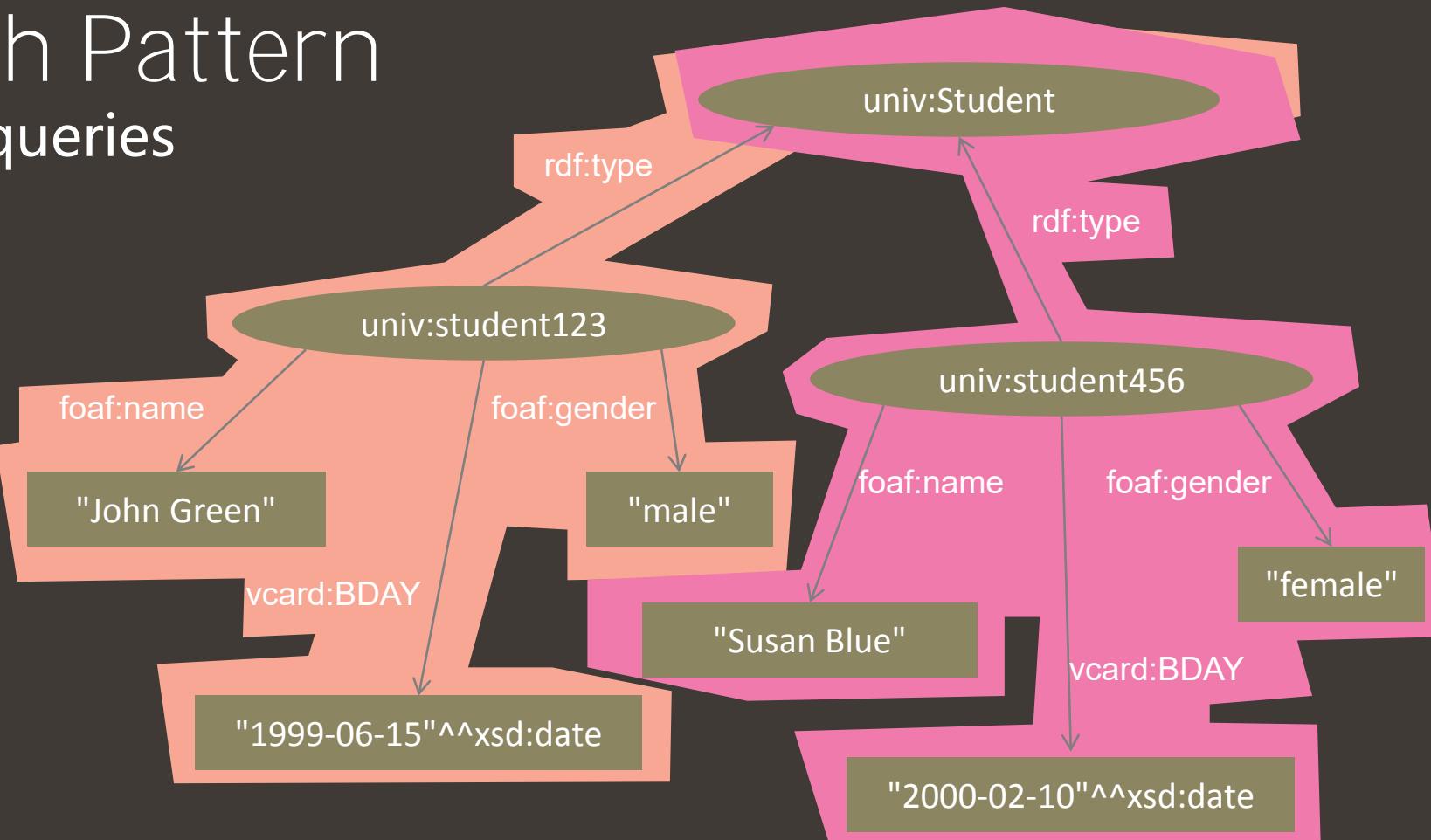
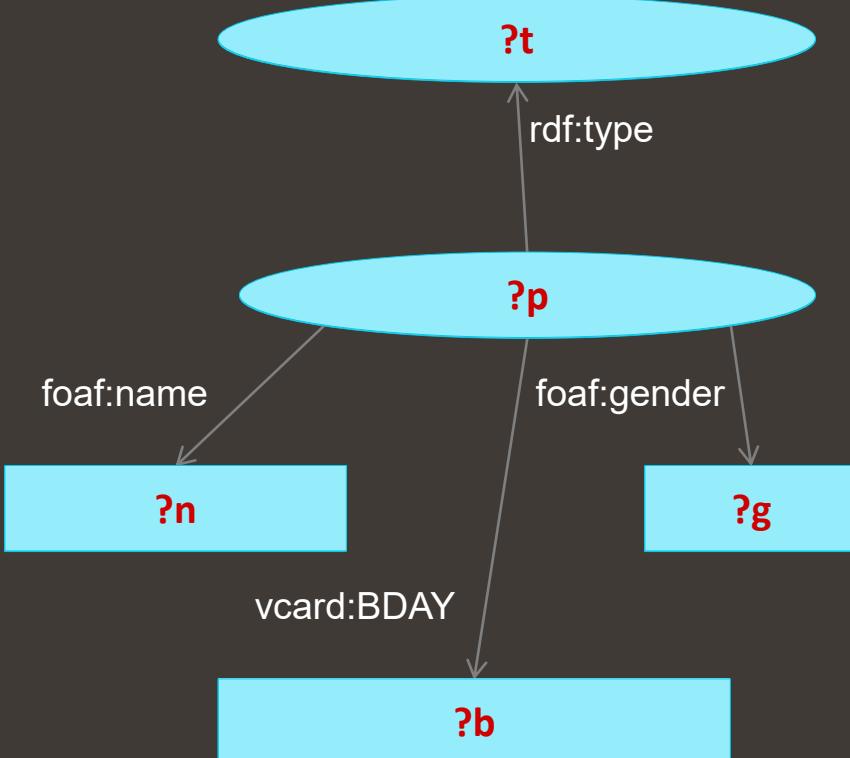


Enterprise capabilities – built on Oracle infrastructure

Manageability, fine-grained security, high availability, integration, and more

SPARQL Graph Pattern

Basic unit of SPARQL queries

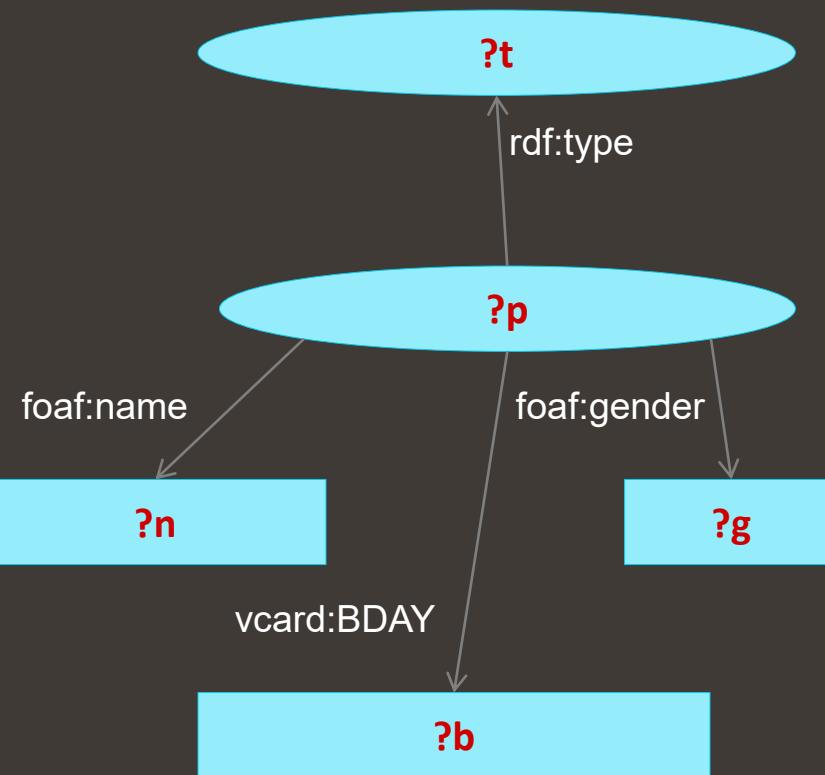


Result 1: {?t=univ:Student, ?p=univ:student123, ?n="John Green", ?g="male", ?b="1999-06-15"^^xsd:date}

Result 2: {?t=univ:Student, ?p=univ:student456, ?n="Susan Blue", ?g="female", ?b="2000-02-10"^^xsd:date}

SPARQL Graph Pattern

Basic unit of SPARQL queries



How do we express this with SPARQL?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT ?t ?n ?b ?g
WHERE
{ ?p rdf:type ?t .
  ?p foaf:name ?n .
  ?p vcard:BDAY ?b .
  ?p foaf:gender ?g }
```

Basic Graph
Pattern (BGP)

SPARQL FILTER: Restricting Solutions

Find all people born before 2000

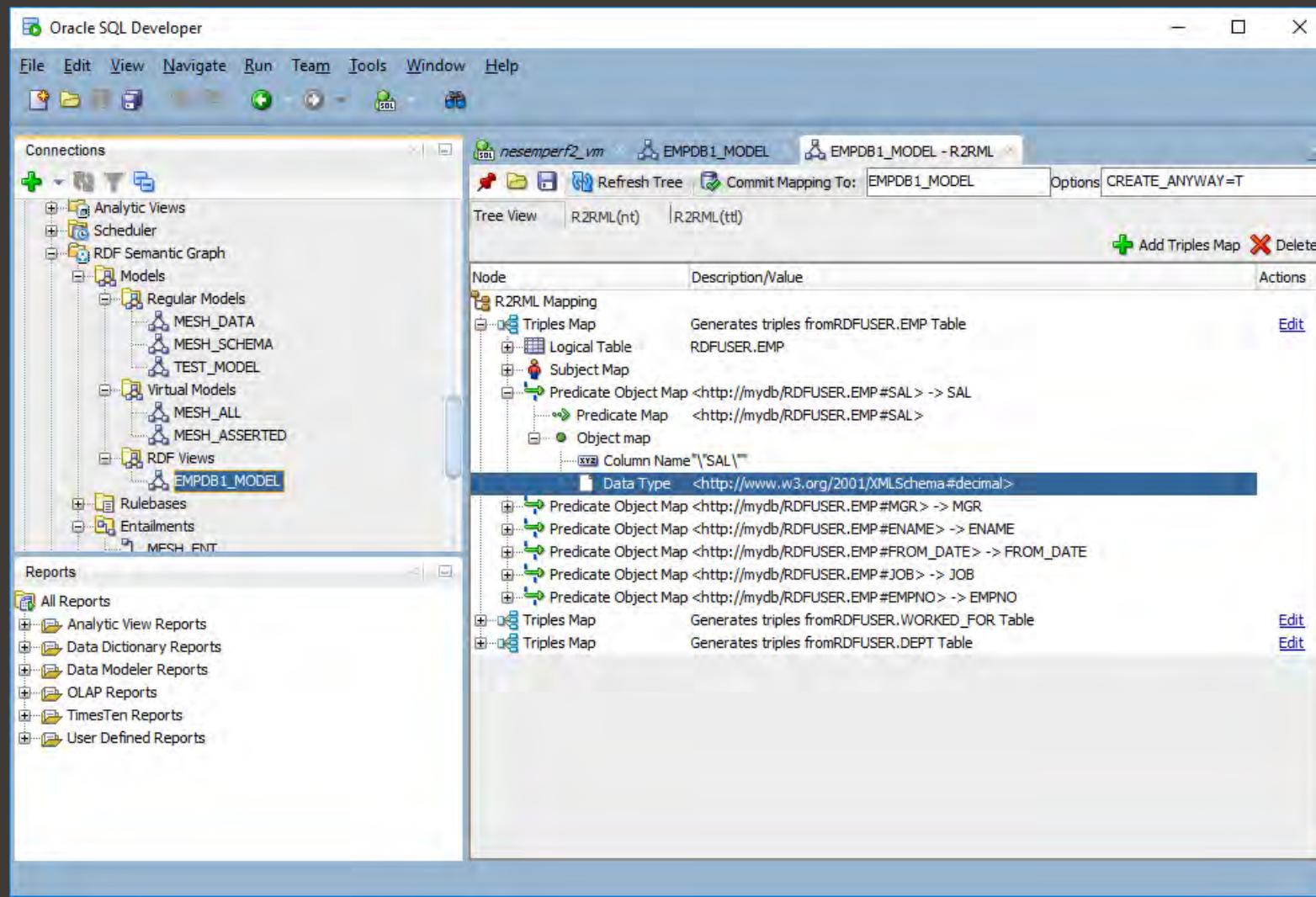
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?n ?b ?g
WHERE
{ ?p foaf:name ?n .
  ?p vcard:BDAY ?b .
  ?p foaf:gender ?g
  FILTER ( ?b < "2000-01-01"^^xsd:date ) }
```

SPARQL Subqueries & Aggregates

Find information about Students that have taken more than 20 classes

```
SELECT DISTINCT ?n ?o ?p ?st ?cnt
WHERE
{ ?s foaf:name ?n .
  ?s vcard:BDAY ?b .
  ?s foaf:gender ?g
  { SELECT ?s (COUNT(?c) AS ?cnt)
    WHERE
      { ?s univ:takes ?c }
    GROUP BY ?s
    HAVING (COUNT(?c) > 20)
  }
}
ORDER BY DESC(?cnt) ASC(?n)
```

Oracle SQL Developer RDF Support



Use Case: Linked Data Publishing

Publishing data in a standard way so that it can be more easily consumed

Popular way for government agencies to publish public data

Why RDF?

URIs

Flexible data model

Standard vocabularies

Standard protocols

The image displays three screenshots of Linked Open Data (LOD) portals, illustrating the use of semantic web technologies for publishing government data:

- Japan National Statistics Center (Screenshot 1):** A screenshot of the LOD portal for the National Statistics Center of Japan. It shows a complex RDF graph visualization with various nodes and relationships, including a bar chart and a line graph. The graph is labeled "Linked Open Data" and "Statistical data of Japan is presented as LOD (Linked Open Data) that is the best rank of open data (5-star open data)".
- Italy National Institute of Statistics (Screenshot 2):** A screenshot of the LOD portal for the National Institute of Statistics of Italy. It features a map of Italy with a red banner at the bottom reading "Dati del Censimento della popolazione e delle abitazioni 2011". Below the map, there are four main sections: "TERRITORIAL LEVEL", "CENSUS VARIABLES", "SPARQL ENDPOINT", and "DATASETS/ONTOLOGIES".
- e-Stat of Japan (Screenshot 3):** A screenshot of the LOD portal for the Statistical Bureau of Japan. It shows a bar chart and a line graph. The page includes a "What's new" section and links to "Outline", "Guidelines", "Providing data", and "RDF Schema".

Japan
National Statistics Center

Use Case: Semantic Data Integration

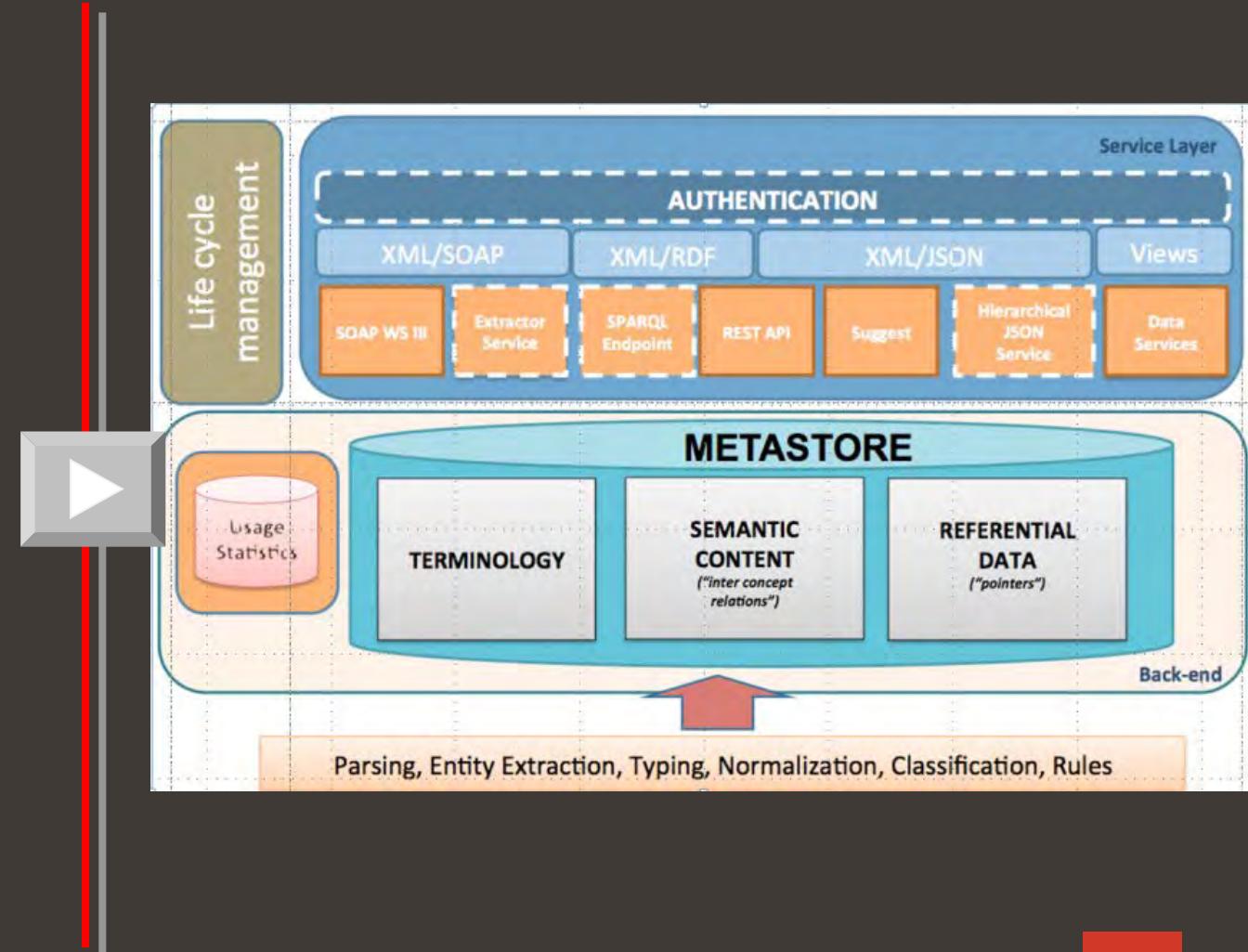
Novartis Institutes for BioMedical Research (NIBR)

Business Challenge

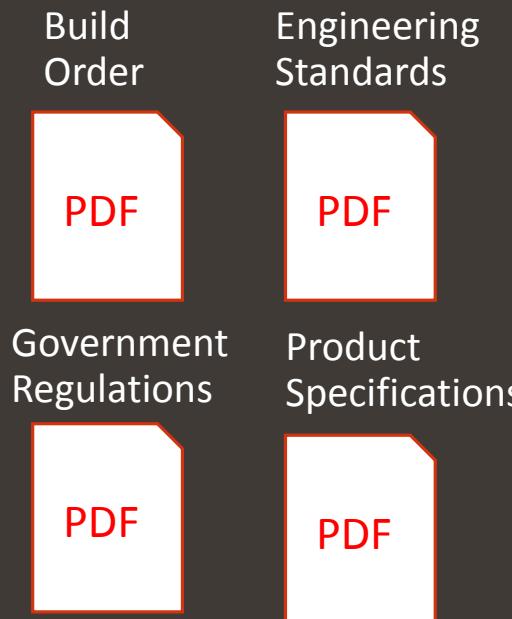
- Link database information on genes, proteins, metabolic pathways, compounds, ligands, etc. to original sources.
- Increase productivity for accessing, sharing, searching, navigating, cross-linking, analyzing internal /external data

Solution

- Semantic integration layer on RDF graph
- Rich domain-specific terminology (biology, chemistry and medicine) 1.6 M terms
- Terminology Hub: 8 GB of referential data that cross-references between data repositories.



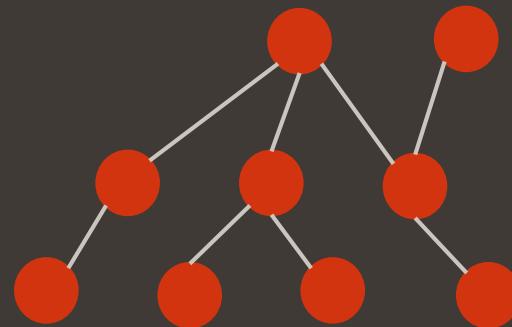
Use Case: Smart Engineering/Construction



Manual assessment by discipline-specific engineers



Requirements and Specifications Ontology



Automated reasoning and queries

Thank You

Melli Annamalai (Melliyal.Annamalai@oracle.com)

Product Manager

Oracle

September 17, 2019

What's Ahead

Tuesday

11:15-12:00 New Tools to Fight Against Financial Crime [CON6222]

Moscone South – Room 204

12:30-1:15 Using Graph Analysis and Fraud Detection in the Fintech Industry
(Paysafe customer session) **Moscone South – Room 152C**

12:30-1:15 Blazing-Fast Distributed Graph Query Processing: 100x as Fast as
Spark [DEV3712] **Moscone South – Room 307**

3:15-4:00 Introducing Oracle Graph Cloud: Automating Graph Analysis
[TRN4754] **Moscone South – Room 159B**

Wednesday

10:00-10:45 Graph Database and Analytics: How To Use Them [TRN 4755]

Moscone South – Room 152C

1:30-2:15 Traversing and Querying Graphs with PGQL and Gremlin with Oracle Spatial
and Graph [DEV4084] **Moscone South – Room 202**

Meet the Experts

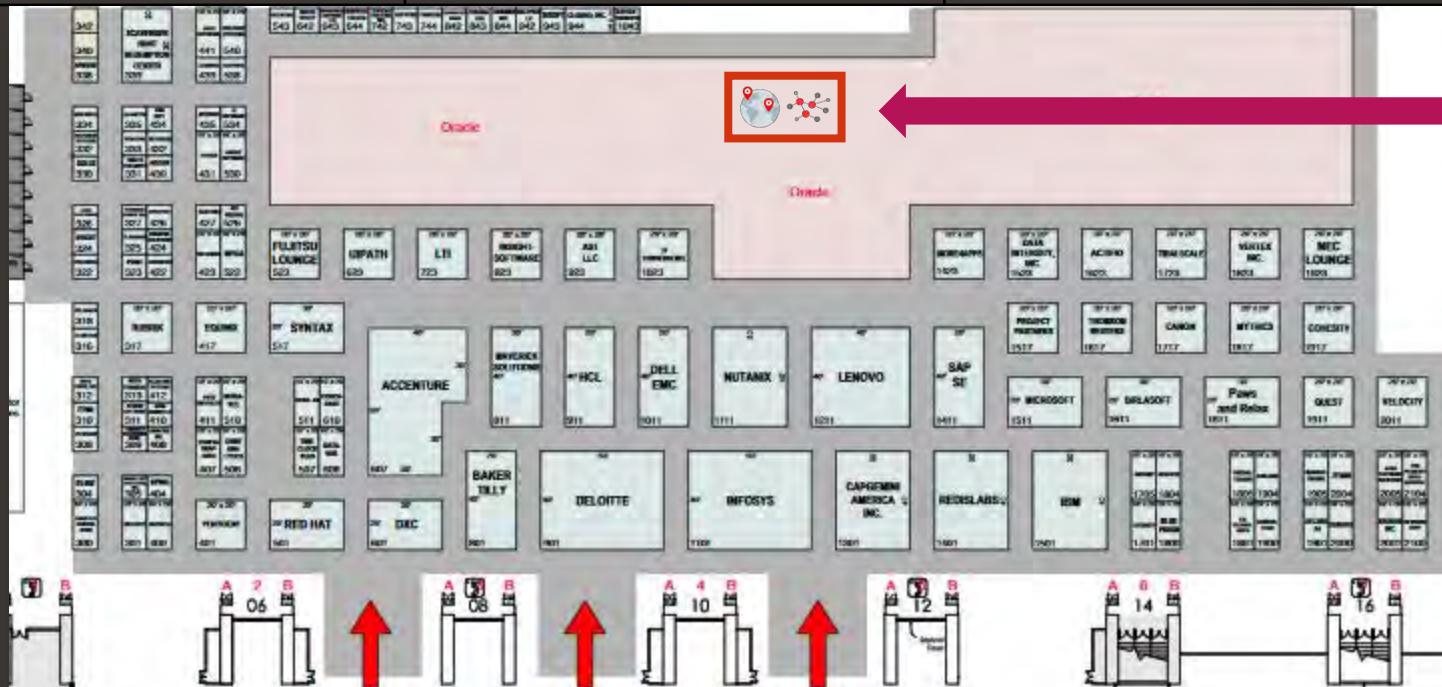
1:30-2:20 Graph Database and Analysis

2:30-3:20 Graph Cloud Service: Automating Graph Analysis

Lounge C, Code One Groundbreakers Hub, Moscone South Level 1

Demogrounds

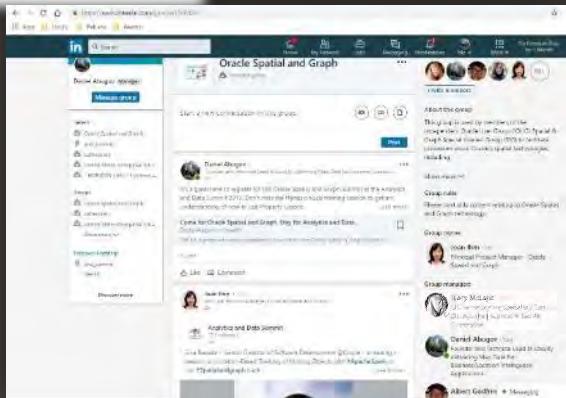
Date/Time	Title	Location
Monday 10:00 am – 4:00 pm		
Tuesday 10:30 am – 5:30 pm		
Wednesday 10:00 am– 4:30 pm	Spatial and Graph: Database, Analytics and Cloud	Moscone South Exhibit Hall ('The Exchange') <ul style="list-style-type: none">• Oracle Demogrounds > Data Management area > Kiosk # ODB-017



The Spatial & Graph SIG User Community

Now part of *BIWA User Group*

We are a vibrant community of customers and partners that connects and exchanges knowledge online, and at conferences and events.



Meet us at OpenWorld! Monday-Wednesday **Moscone West, Level 3, User Group area** at the *BIWA/Analytics Community* table

Join us online
tinyurl.com/oraclespatialcommunity



 oraclespatialsig@gmail.com



SAVE THE DATE

ANALYTICS AND DATA SUMMIT 2020

All Analytics. All Data.
No Nonsense.

February 25-27, 2020

Call for Speakers Now Open!

[SIGN UP FOR OUR NEWSLETTER](#)

Formerly the BIWA Summit with the Spatial and Graph Summit.

@AnalyticAndData



analyticsanddatasummit.org

Seeking customer use cases and technology sessions
Dedicated Spatial & Graph track with 20+ sessions

Session Survey

Help us make the content even better. Please complete the session survey in the Mobile App.

Safe Harbor

The preceding 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.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.