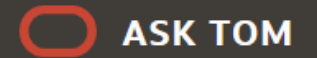


AskTOM Office Hours: Graph Database and Analytics

- Welcome (back) to our AskTOM Graph Office Hours series!
We're back with new product updates, use cases, demos and technical tips
<https://asktom.oracle.com/pls/apex/asktom.search?oh=3084>
- Sessions will be held about once a month
- **Subscribe** at the page above for updates on upcoming session topics & dates
And submit feedback, questions, topic requests, and view past session recordings
- **Note:** **Spatial** now has a new Office Hours series for location analysis & mapping features in Oracle Database:
<https://asktom.oracle.com/pls/apex/asktom.search?oh=7761>

The logo consists of a red circle with a white outline, followed by the text "ASK TOM" in white capital letters on a dark gray background.



Getting Started with Graph Analytics

Melli Annamalai and Ryota Yamanaka

Graph Product Management

Oracle

April 28, 2020



Melli



Nashua, New Hampshire, USA

Ryota



Bangkok, Thailand

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.

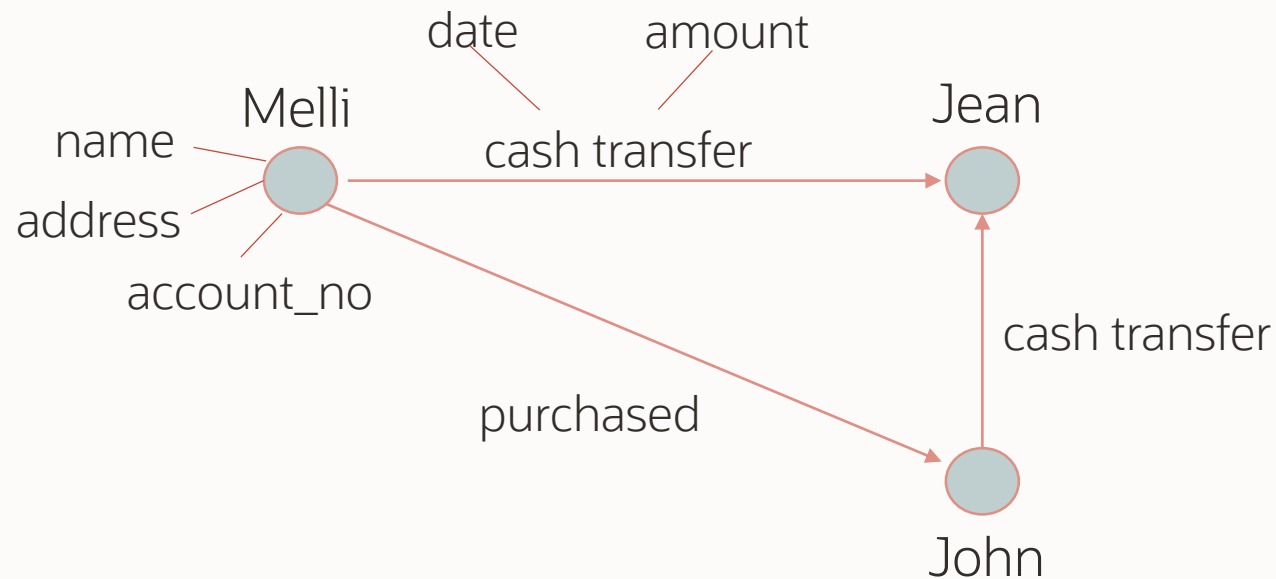
Program Agenda

- Why Graphs?
- Getting Started
 - Modeling your Data as a Graph / Simple PGQL Queries
 - Product Packaging and Shipping
 - Property Graph Architecture Options
- Quick Installation
 - Using **Docker** on Your Machine
 - Oracle Cloud **Always Free** Tier
- Graph Analytics
 - Using Graph **Visualization**
 - Using **Zeppelin** Notebook

Why Graphs?

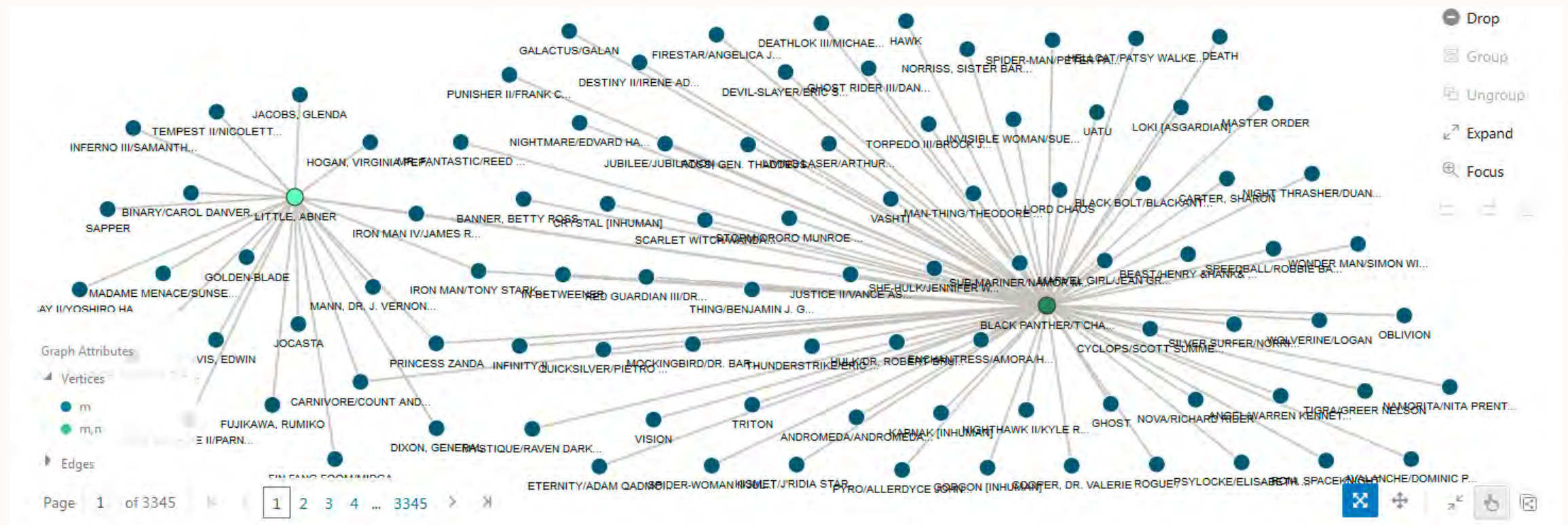
What is a Graph?

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



Finding important nodes: Influencers in a Network

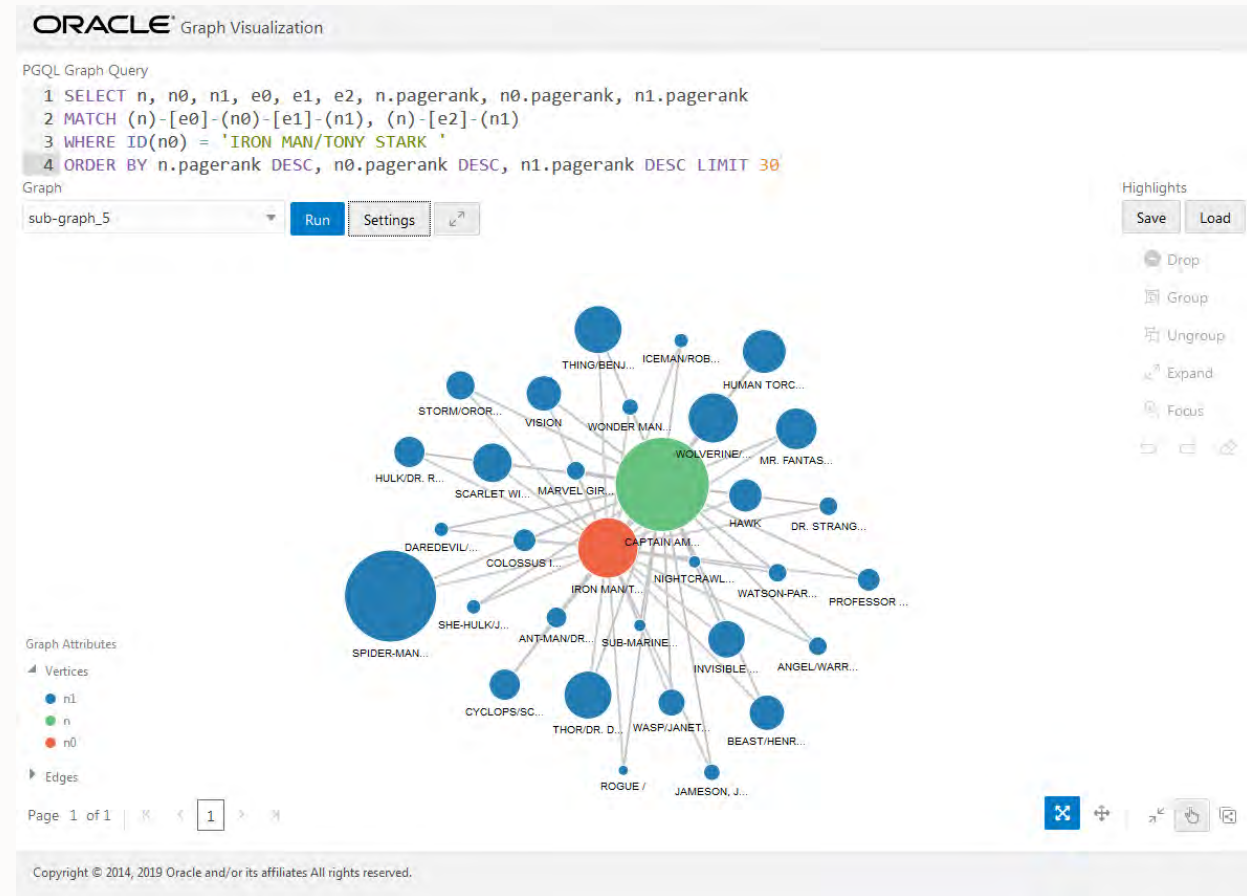
Centrality: Number of edges connected to a node



Finding important nodes: Influencers in a Network

Pagerank:

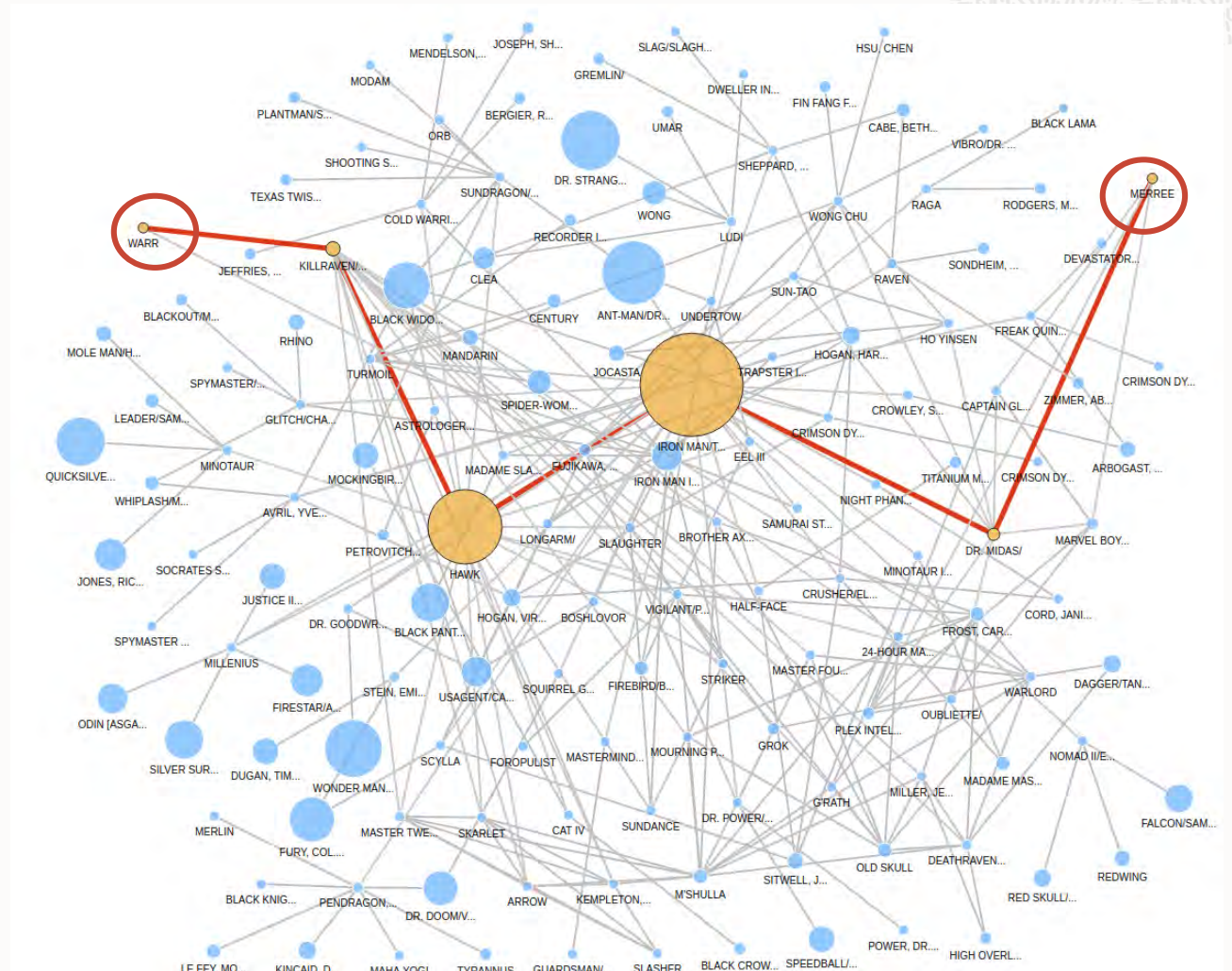
Importance of a node based on nodes connected to it



Path Analysis

Is there a path between two nodes?

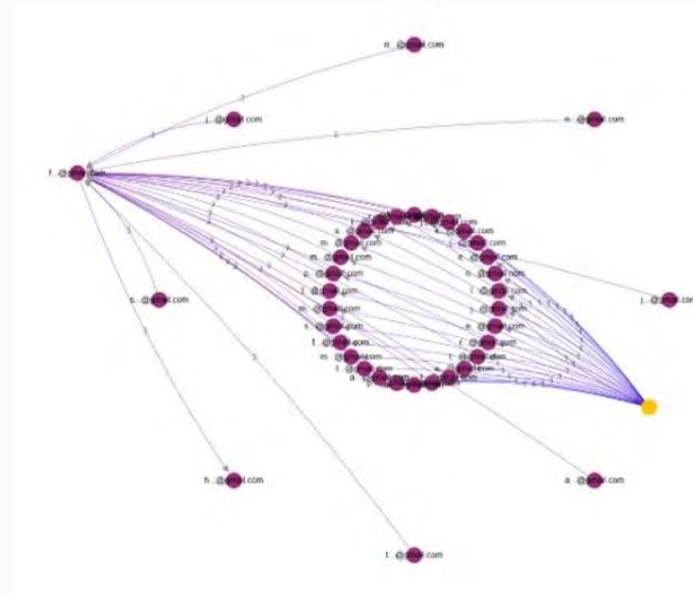
How many hops are there in the shortest path?



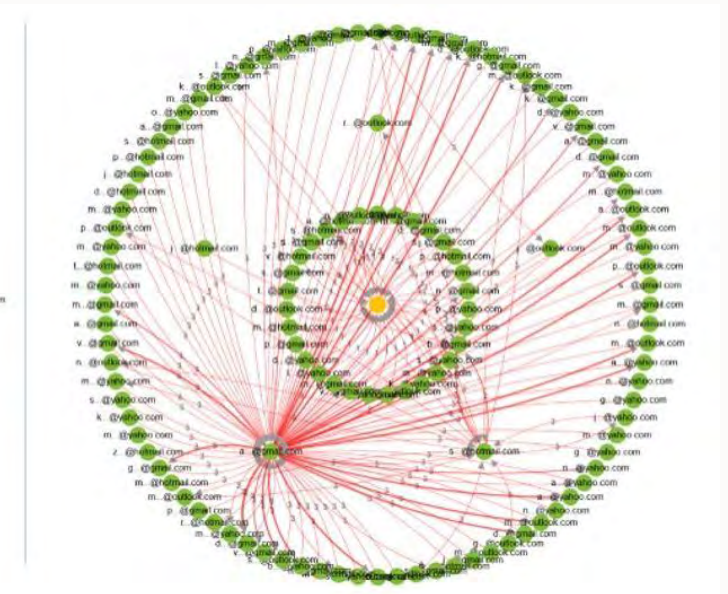
Paysafe: Money Transfer Fraud Detection

Online payment solutions

- Real-time payments, e-wallets
- 1 bn revenue/year
- 500,000 payments/day



Multiple paths going to the same destination



Limited number of source and destination vertices

Property Graph Product Overview



Store, manage, query, and analyze graphs

- **Enterprise capabilities:** Built on Oracle infrastructure
- Manageability, fine-grained security, high availability, integration, and more

Highly scalable

- In-memory query and analytics and in-database query
- 10s of billions of edges and vertices

PGQL: Powerful SQL-like graph query language

Analytics Java API: 50+ pre-built graph analysis algorithms

Visualization

- Light-weight web application, UI accessible from a browser

PGQL Graph Query Language

Graph pattern matching

`(person) - [:works_for] -> (person)`

Basic patterns and reachability patterns

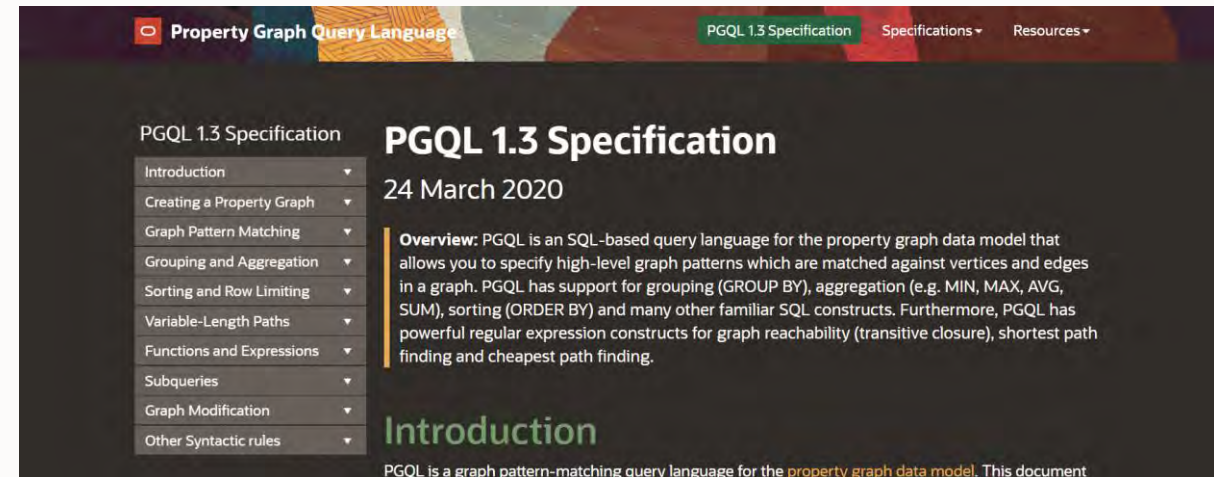
- Can we reach from A to B with an arbitrary number of hops?

Familiarity for SQL users

- Similar language constructs and syntax

`SELECT ... WHERE ...
GROUP BY ... ORDER BY ...`

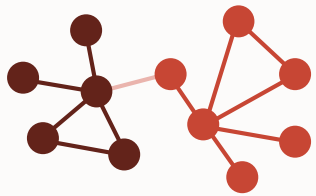
- “Result set” (table) as output



pgql-lang.org

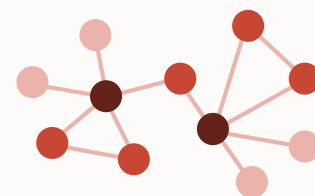
Graph Analytics: 50+ Pre-built Algorithms

Detecting Components and Communities



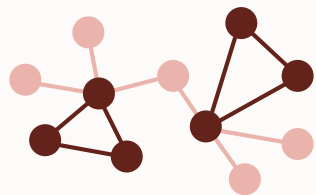
Strongly Connected Components, Weakly Connected Components, Label Propagation, Conductance Minimization,

Ranking and Walking



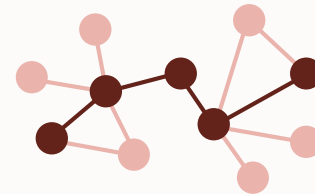
PageRank, Personalized PageRank, Degree Centrality, Closeness Centrality, Vertex Betweenness Centrality, Eigenvector Centrality, HITS, SALSA, Random Walk with Restart

Evaluating Structures



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

Path-Finding



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

Link Prediction

WTF (Who to follow)

Others

Minimum Spanning-Tree, Matrix Factorization

Graph Visualization

ORACLE® Graph Visualization

PGQL Graph Query

```
1 SELECT e
2 MATCH ()-[e]->()
3 LIMIT 100
4
```

Graph

hr

Run

Settings

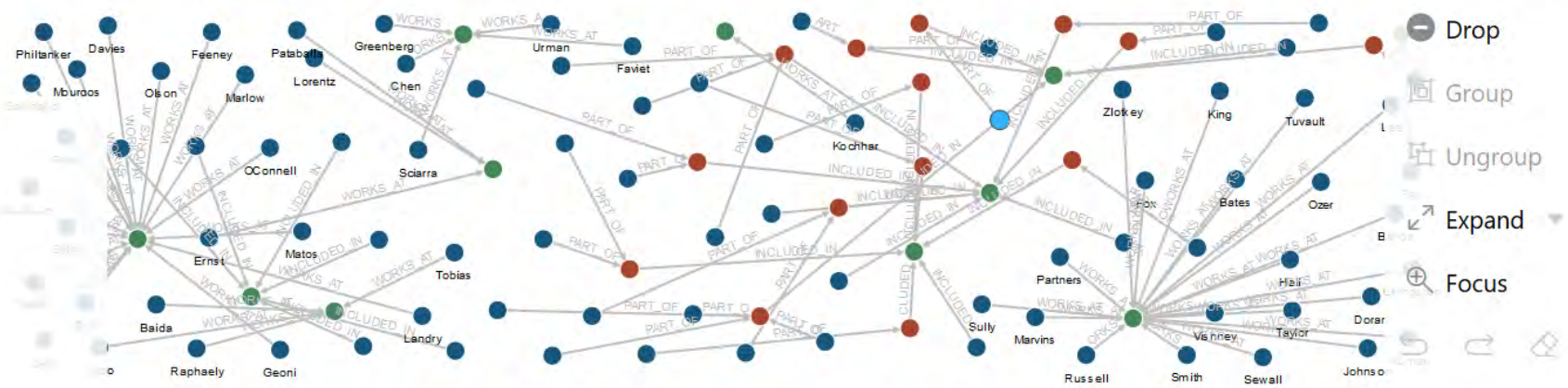
Highlights

Save

Load

Vertices

Edges



Page 1 of 1

1

Drop

Group

Ungroup

Expand

Focus

✕

+

↶

↷

👉

📝

🔍

🔄

Getting Started

Modeling your Data as a Graph

Simple PGQL Queries

Example 1: Modeling Transfer of Money between Bank Accounts

Accounts

	ID	TYPE	ACCOUNT_NO	BALANCE
1	201	account	xxx-yyy-201	1500
2	202	account	xxx-yyy-202	200
3	203	account	xxx-yyy-203	2100
4	204	account	xxx-yyy-204	100
5	211	account	xxx-zzz-204	(null)
6	212	account	xxx-zzz-204	(null)

Customers

	ID	TYPE	NAME	AGE	LOCATION	GENDER	STUDENT
1	101	customer	John	10	Boston	(null)	(null)
2	102	customer	Mary	(null)	(null)	F	(null)
3	103	customer	Jill	(null)	Boston	(null)	(null)
4	104	customer	Todd	(null)	(null)	(null)	true

Cash transfer

	FROM_ID	TO_ID	TYPE	AMOUNT	DATE
1	201	202	transfer	200	2018-10-05
2	211	202	transfer	900	2018-10-06
3	202	212	transfer	850	2018-10-06
4	201	203	transfer	500	2018-10-07
5	203	204	transfer	450	2018-10-08
6	204	201	transfer	400	2018-10-09
7	202	203	transfer	100	2018-10-10
8	202	201	transfer	300	2018-10-10

Account owned by

	FROM_ID	TO_ID	TYPE	SINCE
1	201	101	owned_by	2015-10-04
2	202	102	owned_by	2012-09-13
3	203	103	owned_by	2016-02-04
4	204	104	owned_by	2018-01-05

'owned_by' edge between customer 101 and account 201

- Each table of data entities is a type of vertex
 - Columns are properties of the vertex
- Connections between vertices are edges
 - Foreign key constraints are likely edges
 - Intuitive connections between tables are likely edges

PGQL DDL to Create a Graph

Accounts

ID	TYPE	ACCOUNT_NO	BALANCE
1	201	account xxx-yyy-201	1500
2	202	account xxx-yyy-202	200
3	203	account xxx-yyy-203	2100
4	204	account xxx-yyy-204	100
5	211	account xxx-zzz-204	(null)
6	212	account xxx-zzz-204	(null)

FROM_ID	TO_ID	TYPE	SINCE
1	201	101 owned_by	2015-10-04
2	202	102 owned_by	2012-09-13
3	203	103 owned_by	2016-02-04
4	204	104 owned_by	2018-01-05

Customers

ID	TYPE	NAME	AGE	LOCATION	GENDER	STUDENT
1	101	customer John	10	Boston		
2	102	customer Mary	(null)	(null)		
3	103	customer Jill	(null)	Boston		
4	104	customer Todd	(null)	(null)		

FROM_ID	TO_ID	TYPE	AMOUNT
1	201	202 transfer	200 20
2	211	202 transfer	900 20
3	202	212 transfer	850 20
4	201	203 transfer	500 20
5	203	204 transfer	450 20
6	204	201 transfer	400 20
7	202	203 transfer	100 20
8	202	201 transfer	300 20

PGQL DDL SYNTAX:

```
CREATE PROPERTY GRAPH customer_360
```

```
  VERTEX TABLES (
```

```
    customer PROPERTIES ALL COLUMNS EXCEPT(id)
```

```
    , account PROPERTIES ALL COLUMNS EXCEPT(id)
```

```
  )
```

```
  EDGE TABLES (
```

```
    owned_by
```

```
      SOURCE KEY(from_id) REFERENCES account
```

```
      DESTINATION KEY(to_id) REFERENCES customer
```

```
    , transfer
```

```
      SOURCE KEY(from_id) REFERENCES account
```

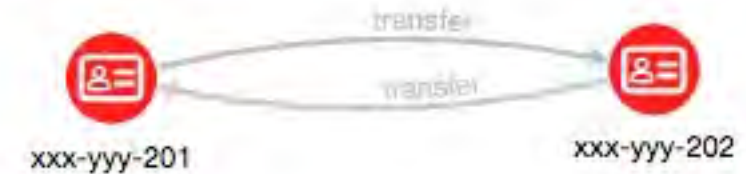
```
      DESTINATION KEY(to_id) REFERENCES account
```

```
  )
```

PGQL Queries on customer_360 Graph

Find circular money transfers

```
SELECT a1.ACCOUNT_NO, t1.DATE, t1.AMOUNT, a2.ACCOUNT_NO, t2.DATE, t2.AMOUNT
FROM customer_360
MATCH (a1)-[t1:TRANSFER]->(a2)-[t2:TRANSFER]->(a1)
WHERE t1.DATE < t2.DATE
```



Find circular money transfers with an additional hop

```
SELECT a1.ACCOUNT_NO, t1.AMOUNT, a2.ACCOUNT_NO, t2.AMOUNT, t3.AMOUNT
FROM customer_360
MATCH (a1)-[t1:TRANSFER]->(a2)-[t2:TRANSFER]->(a3)-[t3:TRANSFER]->(a1)
WHERE t1.DATE < t2.DATE and t2.DATE < t3.DATE
```



AMOUNT

Example 2: Products bought together

PRODUCT_ID	BOUGHT_WITH
0	1
0	2
0	4
1	0
1	12
1	23
...	...

PGQL DDL SYNTAX:

```
CREATE PROPERTY GRAPH products
```

```
VERTEX TABLES (  
    PRODUCTS KEY(PRODUCT_ID) PROPERTIES (PRODUCT_ID)  
)  
EDGE TABLES(  
    SOURCE KEY(PRODUCT_ID) REFERENCES PRODUCTS  
    DESTINATION KEY(BOUGHT_WITH) REFERENCES PRODUCTS  
)
```

- Every product id is a vertex
- Two vertices in one row are connected by an edge - “bought_with” relationship

PGQL Queries

```
-- Find cycles
```

```
SELECT v1.PRODUCT_ID, v2.PRODUCT_ID  
FROM products  
MATCH (v1)-[e1]->(v2)-[e2]->(v1)
```

```
-- Find cycles with a two hop distance
```

```
SELECT v1.PRODUCT_ID, v2.PRODUCT_ID  
FROM products  
MATCH (v1)-[e1]->(v2)-[e2]->(v3)-[e3]->(v1)
```

ORACLE® Graph Visualization

PGQL Graph Query

```
1 SELECT e, e1
2 MATCH (v)-[e]->(v1) -[e1]->(v)
3 LIMIT 10
```

Graph

Run

Settings



Highlights

Save

Load

Drop

Group

Ungroup

Expand

Focus



Vertices



v1

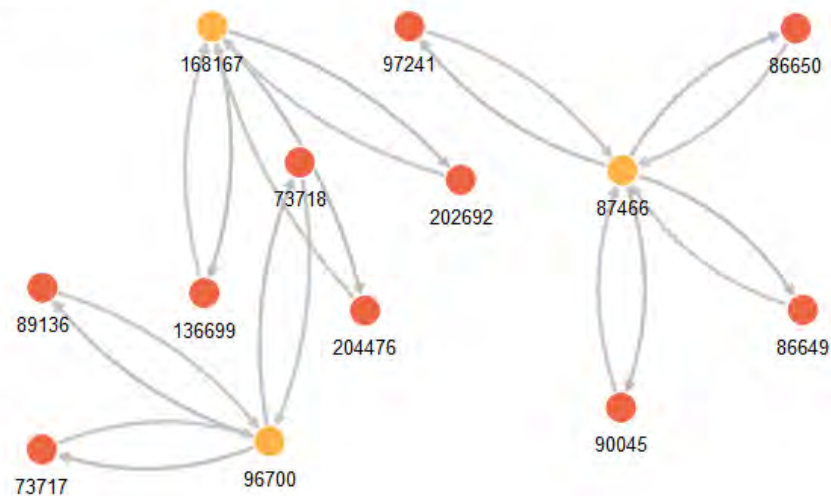


v

Edges



e1



Page 1 of 1



ORACLE® Graph Visualization

PGQL Graph Query

```
1 SELECT e, e1, e2
2 MATCH (v)-[e]->(v1) -[e1]->(v2)-[e2]->(v)
3 LIMIT 10
```

Graph

Run

Settings



Highlights

Save

Load

Drop

Group

Ungroup

Expand

Focus



Vertices



v1,v2



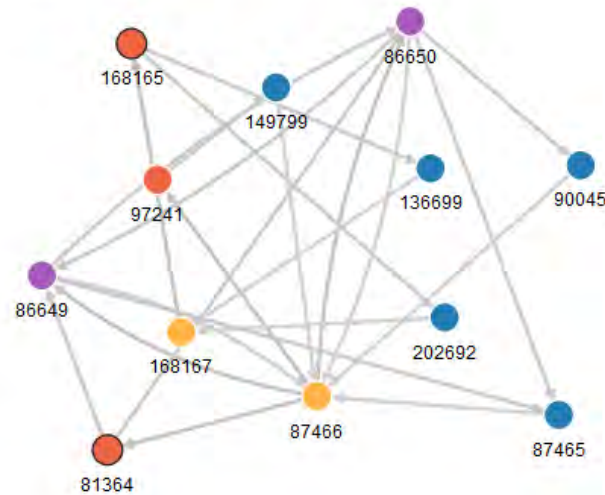
v



v2



v1



Page 1 of 1



1



Graph Analytics Java API



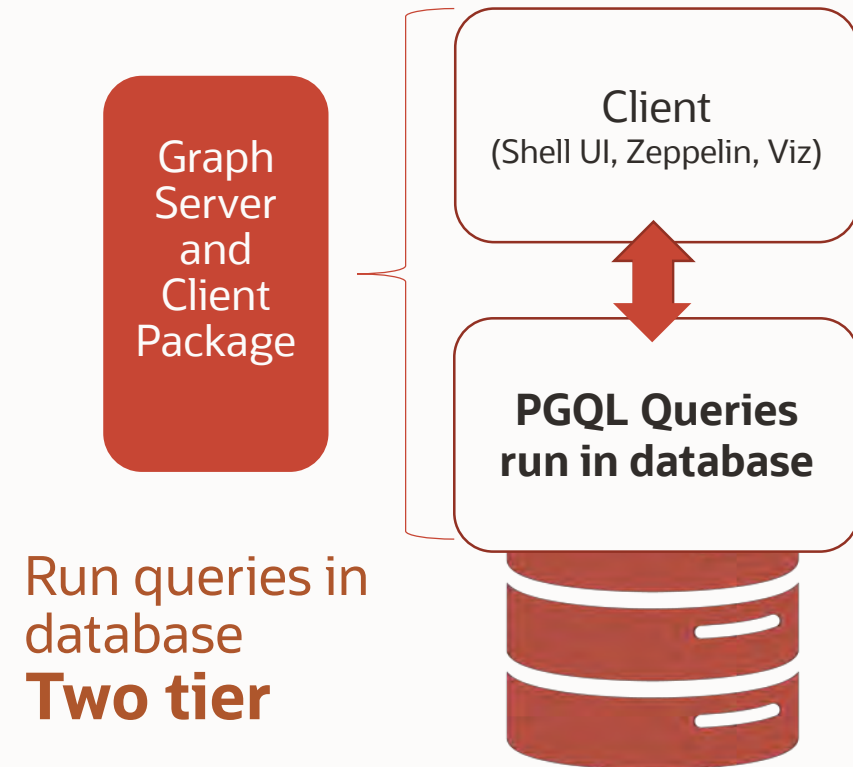
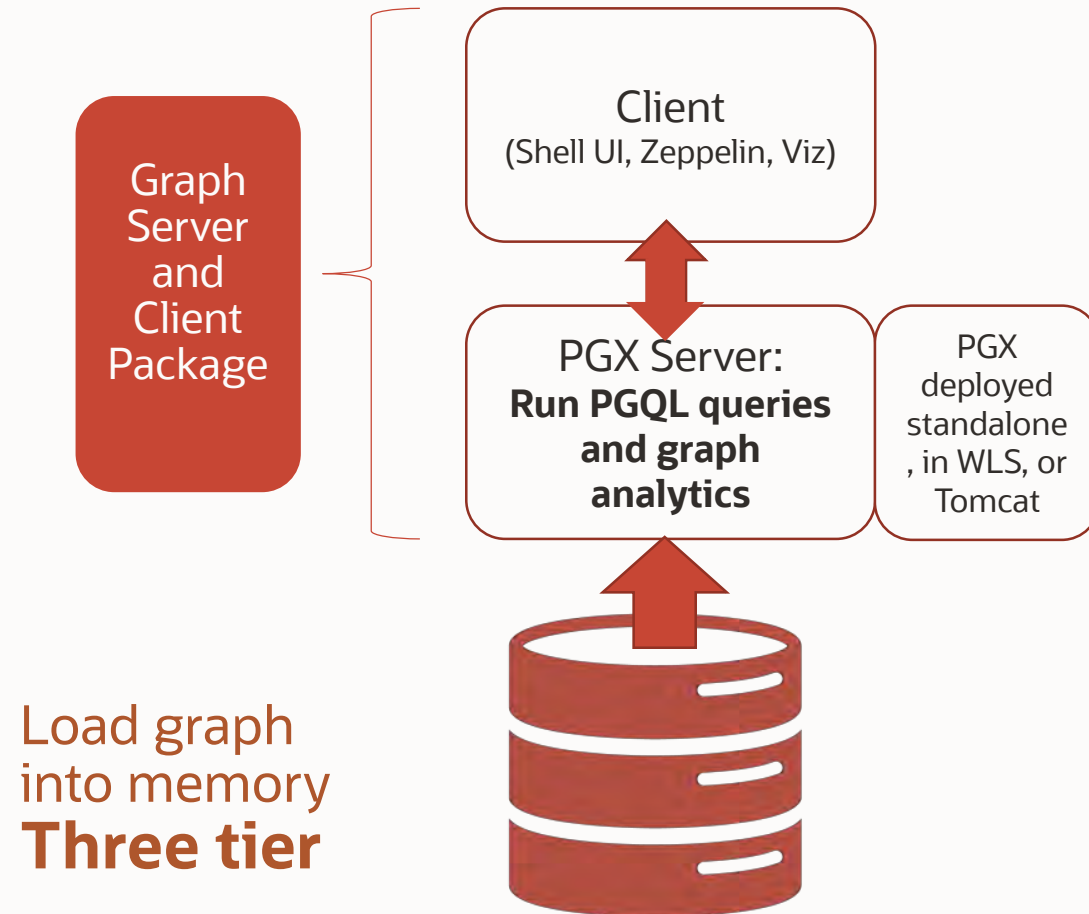
-- use prebuilt algorithms

```
analyst.pagerank(customer_360)
```

```
analyst.betweennessCentrality(customer_360)
```

Property Graph Architecture Options

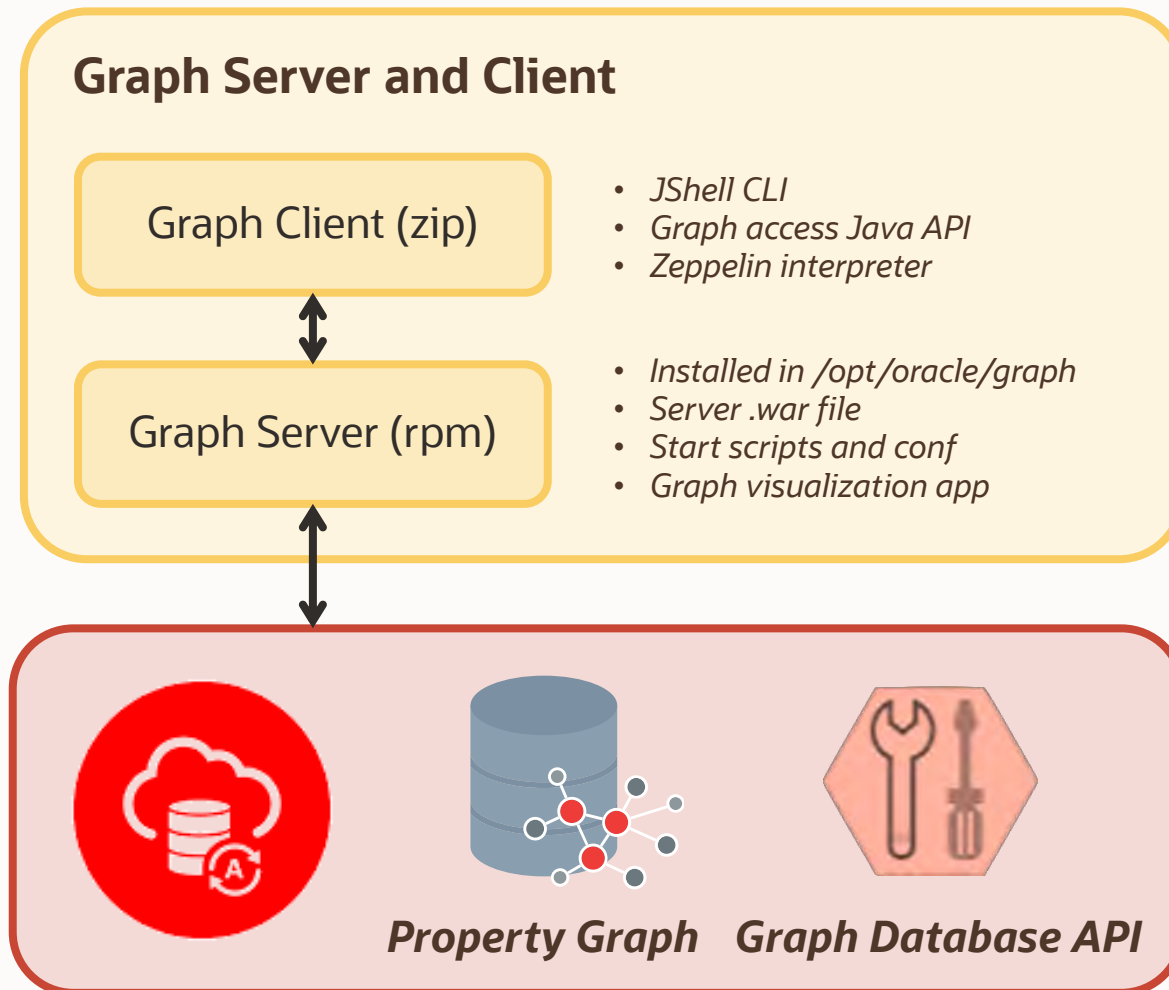
Property Graph Architecture Options





Product Packaging and Shipping

Graph Server and Client



- Graph Server and Client kit
 - Separate download from e-delivery and oracle.com (not shipped with \$ORACLE_HOME)
 - 20.1 released Jan 2020
- Graph Server and Client works with both Database and Big Data

Works with Oracle Database 12.2 onward

- Autonomous Database
 - ADW, ATP
 - ADW free-tier
 - Database Cloud Service
 - Exadata Cloud Service
 - On-prem databases
-
- **Release four times a year**
 - **Easy to uptake latest Graph features**



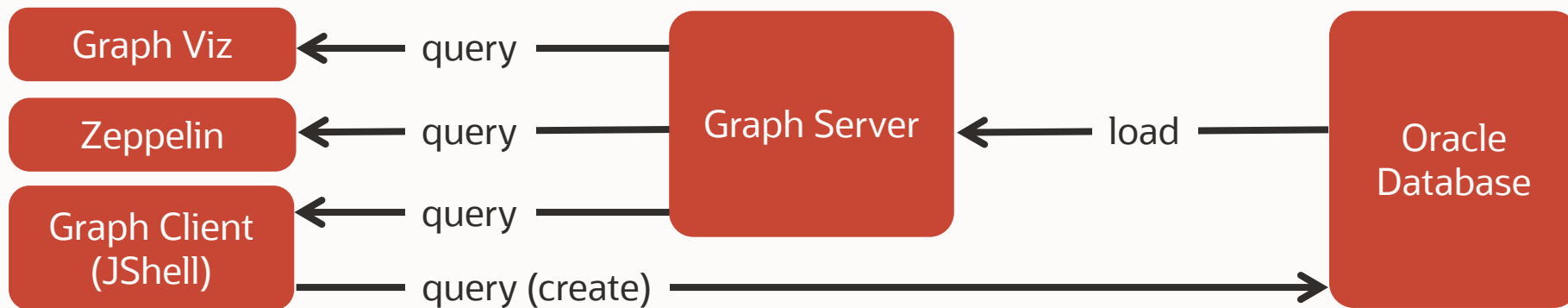


Quick Start

—
Using Docker on Your Machine

Docker Containers

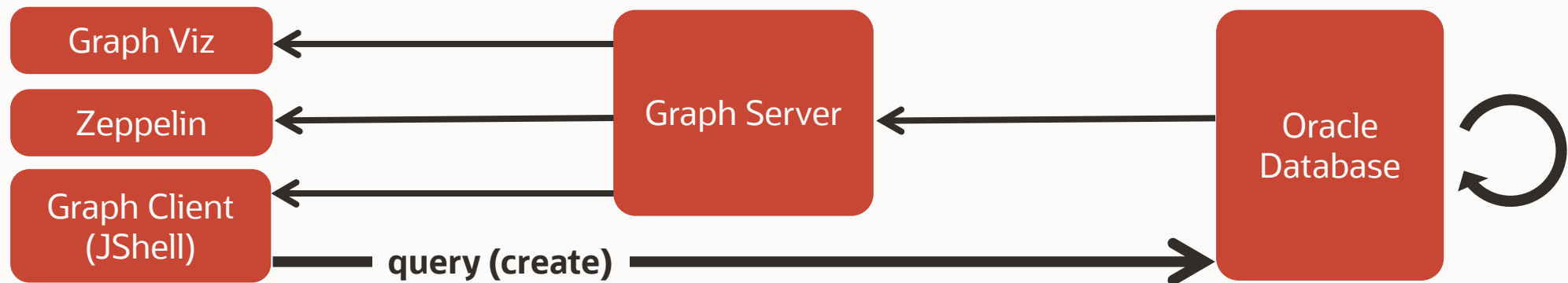
- **Three tier** deployment enables all features (query, algorithm, visualization, notebook)
- For this deployment, 4 containers are composed (Graph Viz works on web browser)



<https://github.com/ryotayamanaka/oracle-pg/wiki>

Create Graph on Database

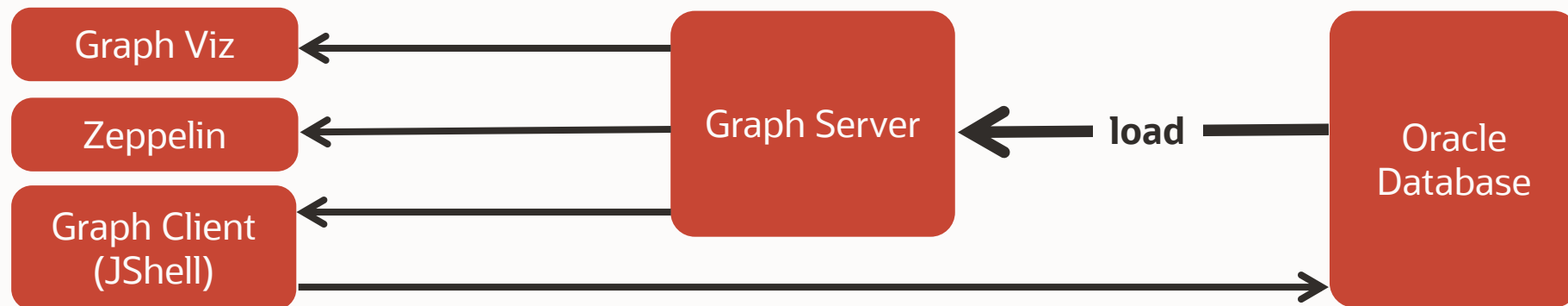
- Table data can be converted to graph data using **CREATE PROPERTY GRAPH** statement
- This statement is issued by Graph Client, and graph data is generated **inside of database**



<https://github.com/ryotayamanaka/oracle-pg/wiki>

Load from Database

- Set the database connection in the loading configuration of this graph.
- Set **preload graphs** to load the graphs when the Graph Server starts.



<https://github.com/ryotayamanaka/oracle-pg/wiki>



Quick Start

—
Oracle Cloud **Always Free** Services

Oracle Cloud "Always Free" Services

What's included with Oracle Cloud Free Tier?

New Always Free

Services you can use for an unlimited time.

- Two Oracle Autonomous Databases with powerful tools like Oracle Application Express (APEX) and Oracle SQL Developer
- Two Oracle Cloud Infrastructure Compute VMs; Block, Object, and Archive Storage; Load Balancer and data egress; Monitoring and Notifications

See below for a list of eligible services



30-day Free Trial

US\$300 in free credits.

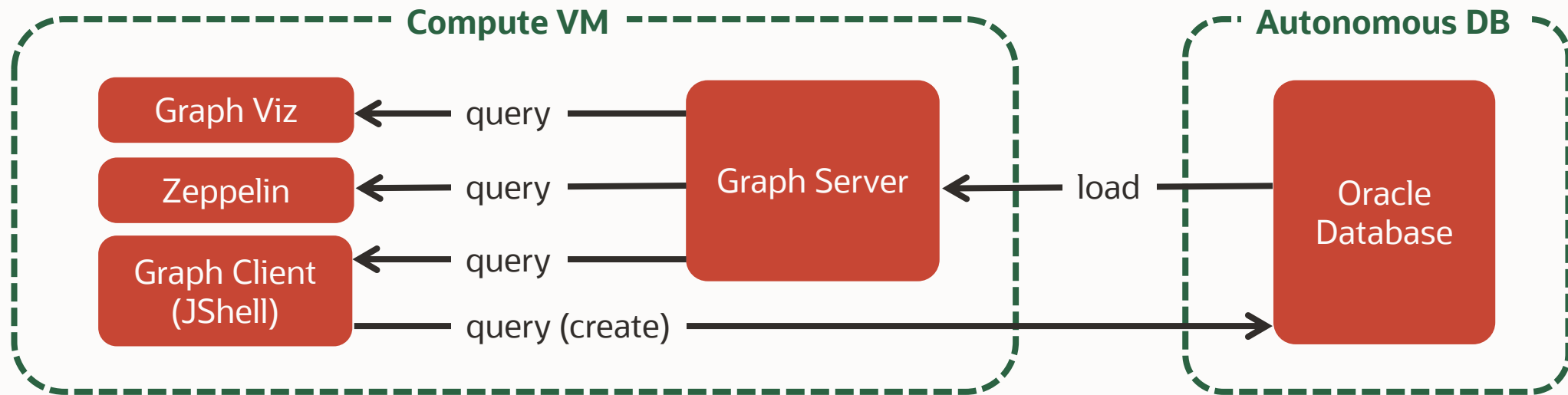
- Access to a wide range of Oracle Cloud services for 30 days, including Databases, Analytics, Compute, and Container Engine for Kubernetes
- Up to eight instances across all available services
- Up to 5 TB of storage

See below for a list of eligible services

<https://www.oracle.com/cloud/free/>

"Always Free" Graph Analytics Platform

- Always Free Services include 2 Compute VMs and 2 Autonomous Databases
- Using 1 of each, we can build an "always free" graph analytics platform



<https://github.com/ryotayamanaka/oracle-pg/wiki>



Graph Analytics

Using Graph Visualization

Customer 360 Analysis

In customer 360 degrees analysis, **data integration** is the first step. This small graph data is assumed to be an integrated dataset of the followings in financial industry:

- **accounts** and account owners
- **transactions** between these accounts
- **purchases** by the people who own the accounts



<https://github.com/ryotayamanaka/oracle-pg/wiki/Customer-360-Analysis>

Customer 360 Analysis

In typical graph visualization, users can select nodes and **"expand"** them to pull all related information for specific entities, for knowing or investigating more about their customers.

```
SELECT a
MATCH (a:Account)
WHERE a.account_no = 'xxx-yyy-201'
```

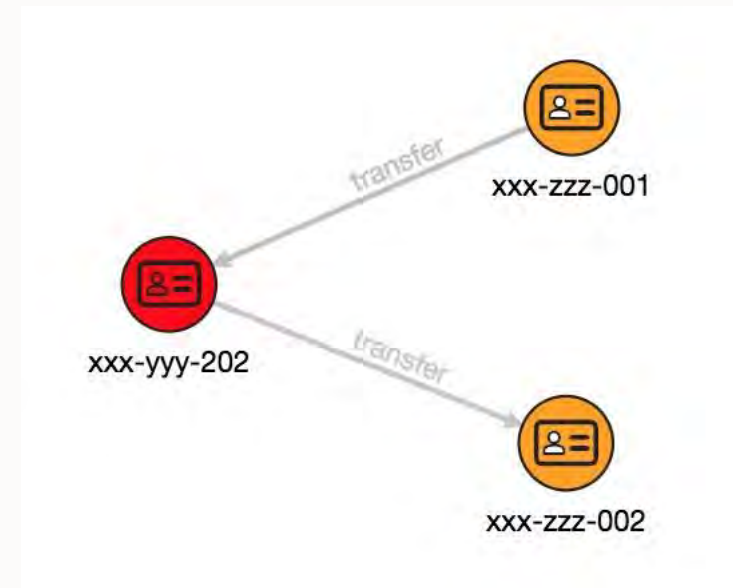


Pattern Matching

PGQL Query is convenient for detecting specific patterns.

Find accounts that had an inbound and an outbound transfer, of over 500, on the same day, and its balance is less than 300

```
SELECT
  a.account_no, a.balance,
  t1.amount, t2.amount, t1.date
MATCH (a)<-[t1:transfer]-(a1)
      , (a)-[t2:transfer]->(a2)
WHERE t1.date = t2.date
      AND t1.amount > 500
      AND t2.amount > 500
      AND a.balance < 300
```

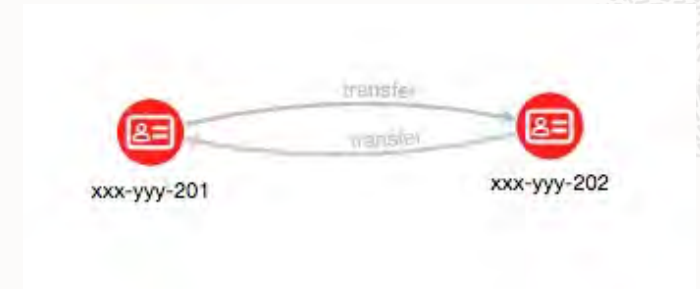


Detection of Cycles

Find a series of transfers that start and end at the same account such as A to B to A, or A to B to C to A

```
SELECT
  a1.account_no, t1.date, t1.amount,
  a2.account_no, t2.date, t2.amount
MATCH (a1)-[t1:transfer]->(a2)-[t2:transfer]->(a1)
WHERE t1.date < t2.date
```

```
SELECT
  a1.account_no, t1.amount, a2.account_no,
  t2.amount, a3.account_no, t3.amount
MATCH (a1)-[t1:transfer]->(a2)-[t2:transfer]->(a3)-[t3:transfer]->(a1)
WHERE t1.date < t2.date
      AND t2.date < t3.date
```



Graph Analytics

—
Using Zeppelin Notebook

Influential Accounts

Which is the account with the highest pagerank?

Filter by "transfer" relationship, run **pagerank** algorithm. The result is stored as "pagerank" new node property, and it can be queried.

```
sg = g.filter(new EdgeFilter(
    "edge.label()='transfer'"));
```

```
analyst.pagerank(sg);
```

```
SELECT a.account_no, a.pagerank
MATCH (a)
ORDER BY a.pagerank DESC
```



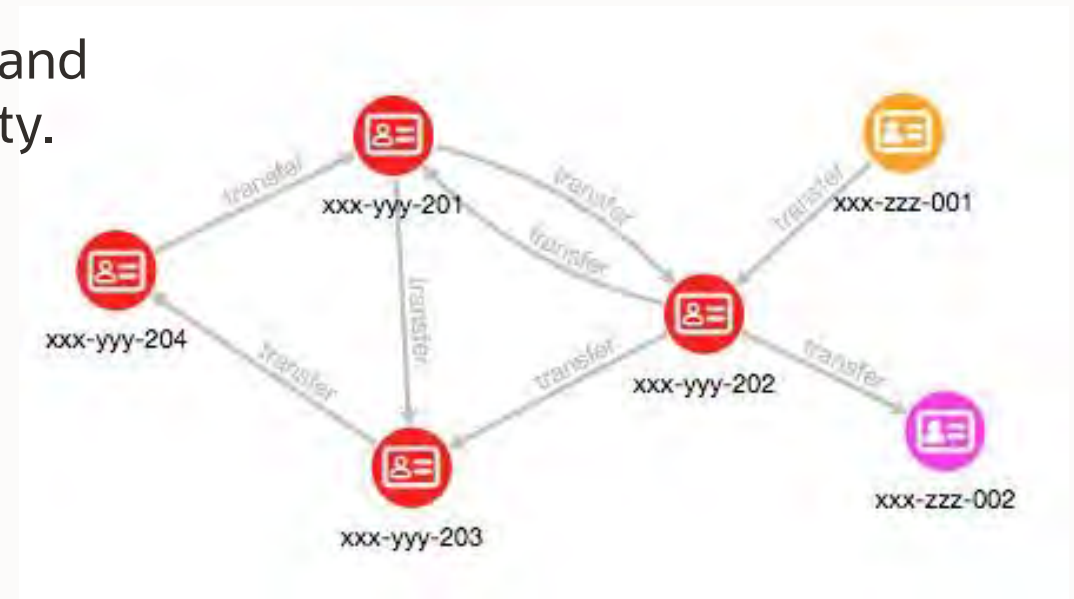
Community Detection

Find the communities where every account is reachable from every other account

Run **strongly connected component** algorithm and reflect the results into "component" node property.

```
result = analyst.sccKosaraju(sg)
```

```
SELECT  
  a.component, COUNT(a.account_no)  
MATCH (a)  
GROUP BY a.component
```

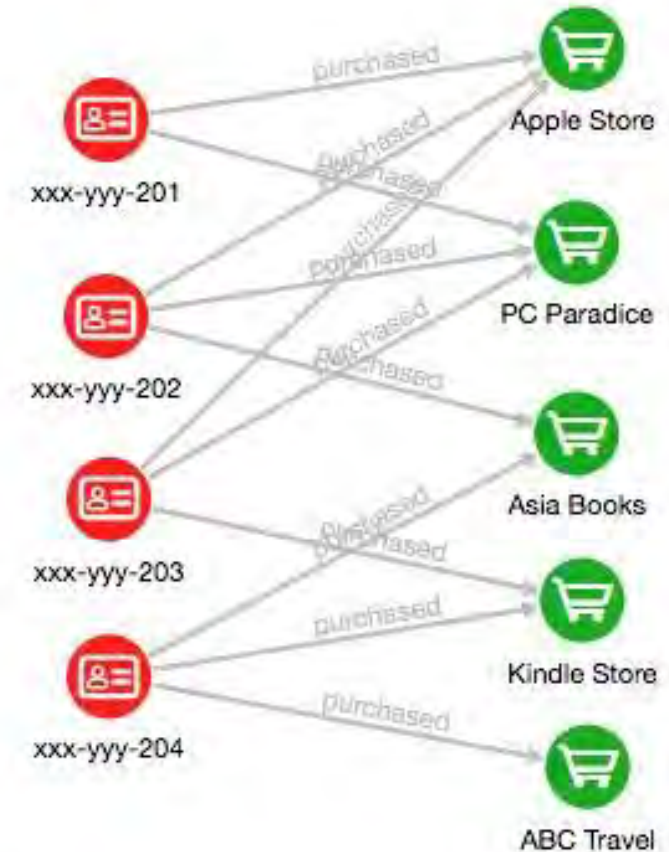


Recommendation

Which merchants should be recommended to the account "-201" based on the purchase history?

Filter by "purchased" relationship, run **personalized pagerank** from the node 201, and then query :

```
SELECT ID(x), x.name, x.pagerank
MATCH (x)
WHERE x.type = 'merchant'
AND NOT EXISTS (
  SELECT *
  MATCH (x)-[:purchased_by]->(a)
  WHERE ID(a) = 201
)
ORDER BY x.pagerank DESC
```





Further References

Partial List of Planned Features

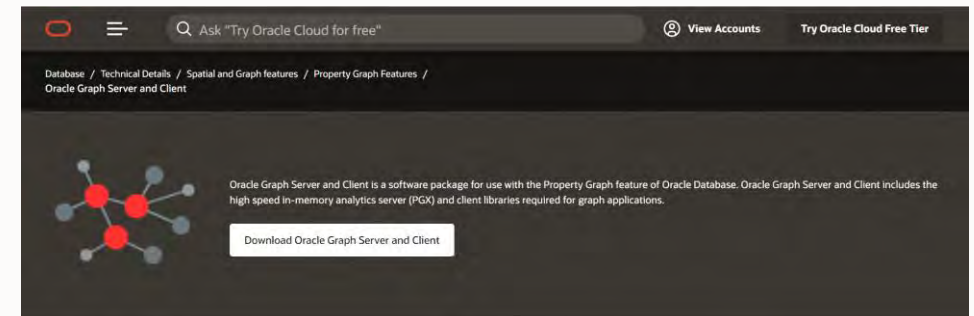


- Graph Server and Client on OCI Marketplace
- PGQL INSERT-UPDATE-DELETE
- Improved delta refresh capabilities
- Support for GraphViz when running PGQL in the database
- Use SQL CL and SQL Developer to run PGQL queries in the database
- Native Python client
- SQL/PGQ support

Helpful Links

- Graphs at Oracle
<https://www.oracle.com/goto/graph>
- Oracle Property Graph
<http://www.oracle.com/goto/propertygraph>
- Blog: Examples, Tips and Tricks
<http://bit.ly/OracleGraphBlog>
- AskTOM Series: <https://asktom.oracle.com/pls/apex/asktom.search?office=3084>
- Social Media
 - Twitter: @OracleBigData, @SpatialHannes, @Jeanlhm
 - LinkedIn: Oracle Spatial and Graph Group
 - YouTube: youtube.com/c/OracleSpatialandGraph

Search for 'Oracle Graph Server and Client' to [download](#) from oracle.com



The Spatial & Graph User Community

- A part of [Analytics and Data Oracle User Community](#) (formally BIWA)
- Vibrant community of tech enthusiasts including customers, partners, students
- We share knowledge online, and at conferences and events
- Global – Americas, Europe, Africa, Asia

Join us



LinkedIn Oracle Spatial and Graph group
[linkedin.com/groups/1848520/](https://www.linkedin.com/groups/1848520/)



@oraspatialsig

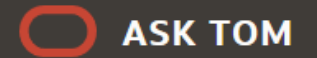


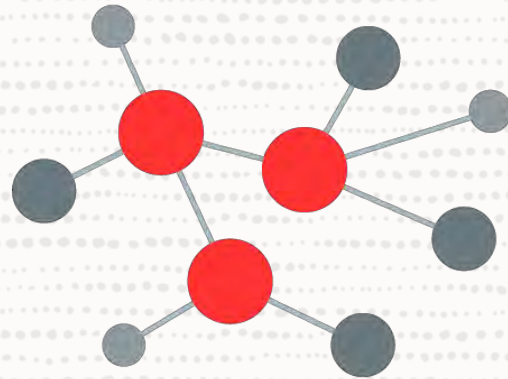
oraclespatialsig@gmail.com



AskTOM Office Hours: Graph Database and Analytics

- Today's session will be repeated on April 30 for APAC (8:30 India | 10:00 Bangkok | 12:00 Tokyo | 13:00 Sydney)
- Next session - save the date:
 - **Topic:** Financial Industry Use Cases for Graph Analytics
 - **Date:** May 27 & 28
Check back at landing page for details
- Recording of today's session will be available at the landing page
- **Subscribe** for updates on upcoming session topics & dates
- Submit feedback, questions, topic requests, and view past session recordings
<https://asktom.oracle.com/pls/apex/asktom.search?oh=3084>

The logo consists of a red circle with a white outline, followed by the text "ASK TOM" in white capital letters on a dark background.



Thanks for attending! See you next time.

<https://asktom.oracle.com/pls/apex/asktom.search?oh=3084>