

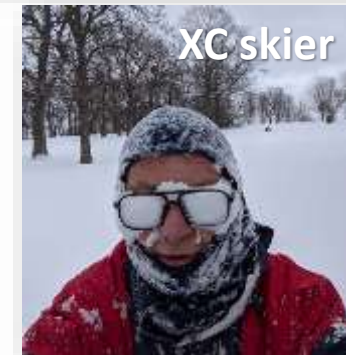
# Grafting Grifters: Identify & Display Patterns of Corruption With Oracle Graph



**2022 Webinar Series**  
**March 8, 2022**

**Jim Czuprynski**  
**@JimTheWhyGuy**  
**Zero Defect Computing, Inc.**

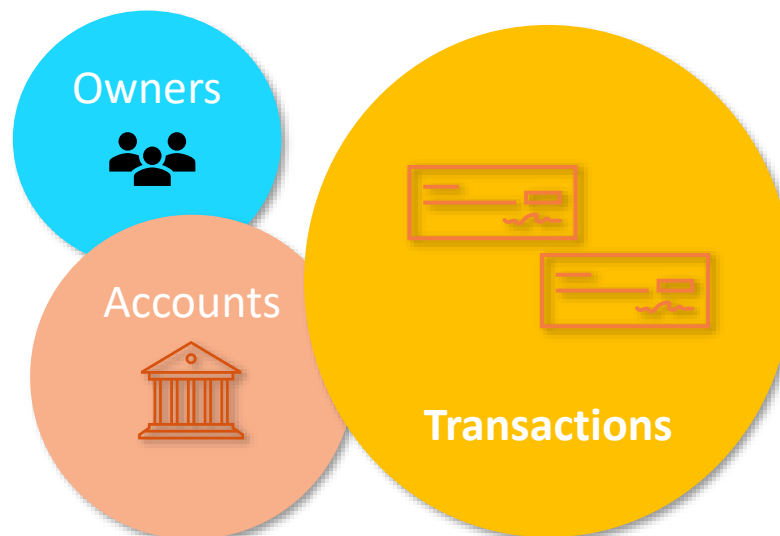
# Who Am I, and What Am I Doing Here?



- E-mail me at [jim@jimthewhyguy.com](mailto:jim@jimthewhyguy.com)
- Follow me on Twitter (@JimTheWhyGuy)
- Connect with me on LinkedIn (Jim Czaprynski)

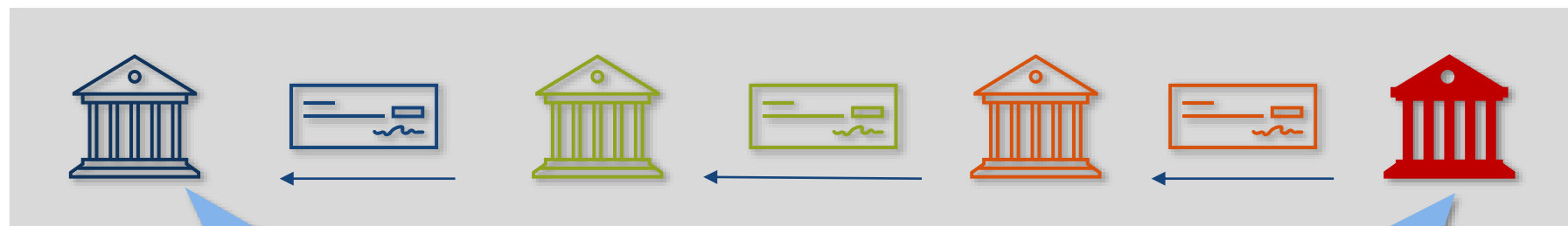
# Property Graphs: Finding Patterns *Between* Data Elements

Traditional RDBMS databases use SQL in a **set-based** fashion



This makes it **easy** to find result sets with data **in common** (or **not** in common) ...

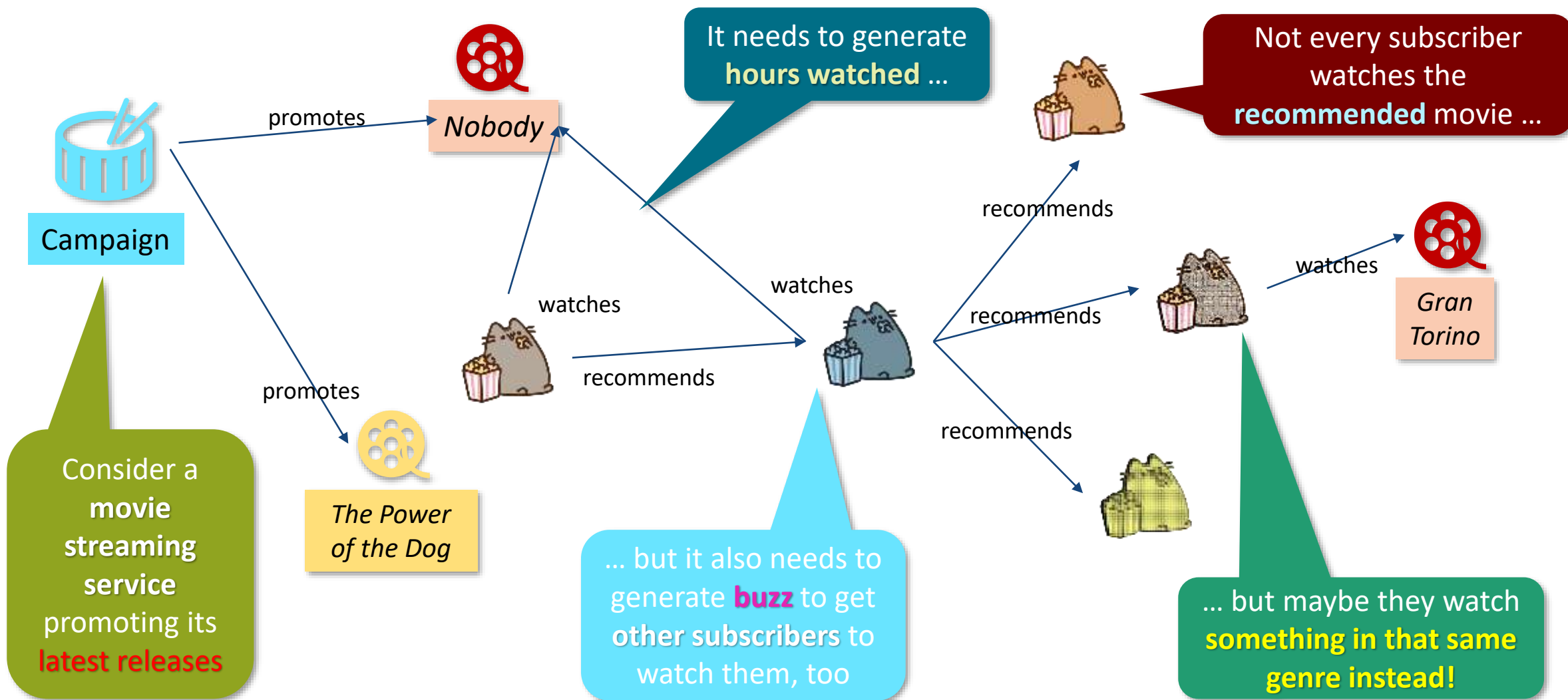
... but it makes it ***much harder*** to identify find patterns of how data is **linked together**



Given a particular **bank account** as a starting point, has that account's **owner** ...

... received funds from **another** bank account that's owned by someone **in a different country?**

# Property Graphs: Not About Data *Itself*, But How They're *Connected*





# Identifying Corruption Via MOE (Mark One Eyeball)

	A	B	C	D	E	F
1	owner_id	name	address	city	state	country
2	166	Vladimir Ivan Ivanovich		Moscow		
3	266	Jakub Wojciechowski	579 Allen Rd	Krakow		
4	366	Fraud Assurance	37 West 57th Street	New York	NY	
5	466	Bordeaux Oenoophilia GmbH	1000 Pennsylvania Avenue	Washington	DC	
6	566	Cotroceni Civic Holdings	1000 Pennsylvania Avenue	Washington	DC	

	A	B	C	D	E	F
1	src_acct_id	tgt_acct_id	tx_date	tx_amount	tx_type	tx_memo
2	50566	10166	2021-02-01	8500000	RFW	Investment withdrawal
3	50566	10166	2021-01-29	6500000	RFW	Investment withdrawal
4	98466	50566	2021-01-31	3250000	RFC	Investment funding
5	98466	50566	2021-01-30	2750000	RFC	Investment funding
6	98466	50566	2021-01-29	2500000	RFC	Investment funding
7	40466	50566	2021-01-28	2717068	RFC	Investment funding
8	40466	50566	2021-01-23	3782932	RFC	Investment funding
9	10166	98166	2020-11-12	15000000	RFC	Investment funding
10	98366	40466	2020-12-27	4500000	RES	Property sales
11	98366	40466	2020-12-23	1500000	RES	Apartment sale
12	30366	98366	2021-01-22	8000000	REP	Parking garage purchase
13	30366	98366	2020-11-18	7000000	REP	Apartment purchase
14	98166	30366	2021-01-21	2500000	CON	Consulting services
15	98166	30366	2020-12-29	7500000	CON	Consulting services
16	98166	30366	2020-11-14	5000000	CON	Consulting services
17	98366	98466	2021-01-28	4500000	CHR	Donation
18	98366	98466	2021-01-24	500000	CHR	Donation
19	98366	98466	2021-01-23	1250000	CHR	Donation
20	98366	98466	2021-01-22	1250000	CHR	Donation
21	98366	40466	2020-12-28	500000	CHR	Donation
22	98366	98466	2020-12-25	1000000	CHR	Donation

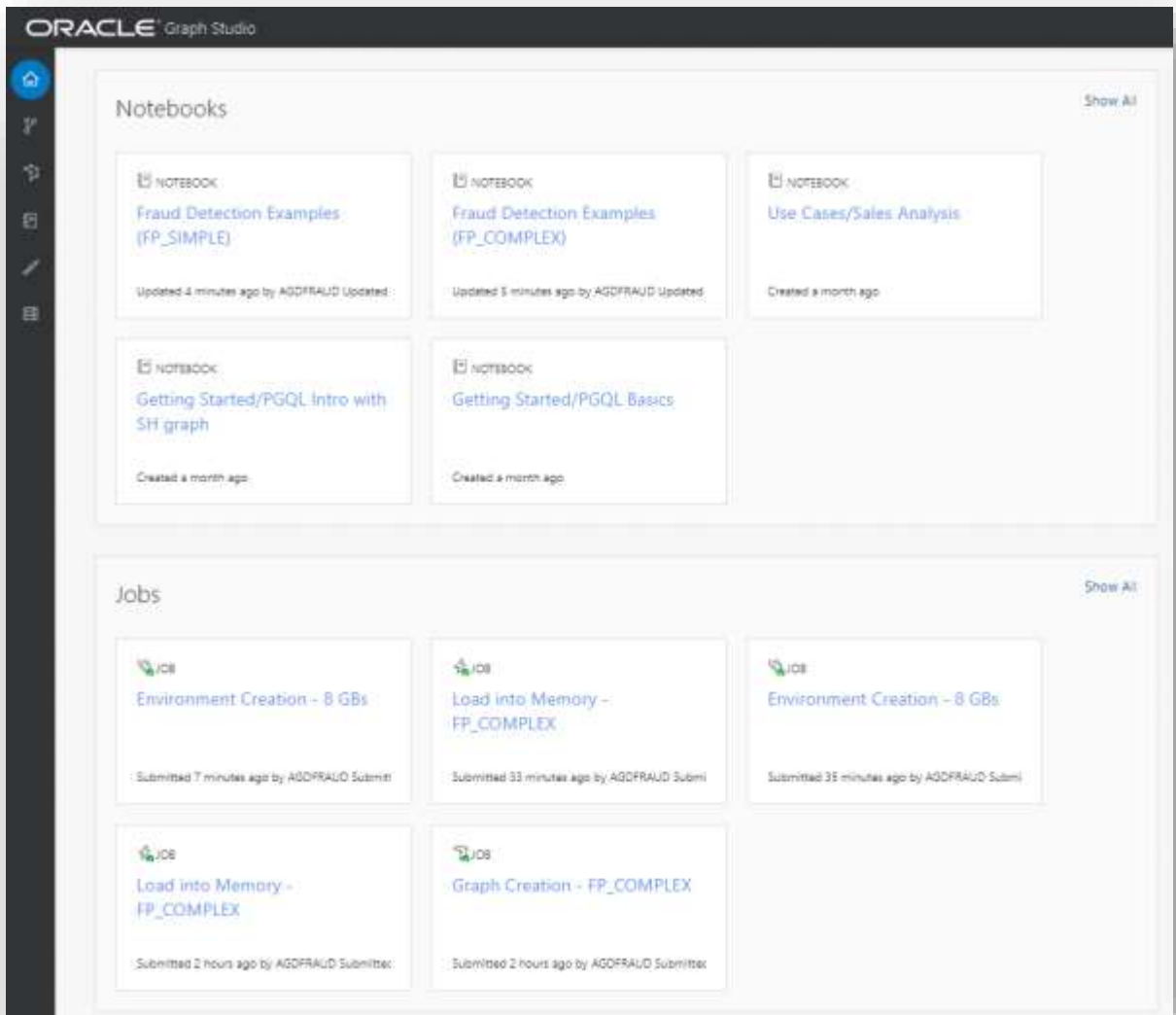
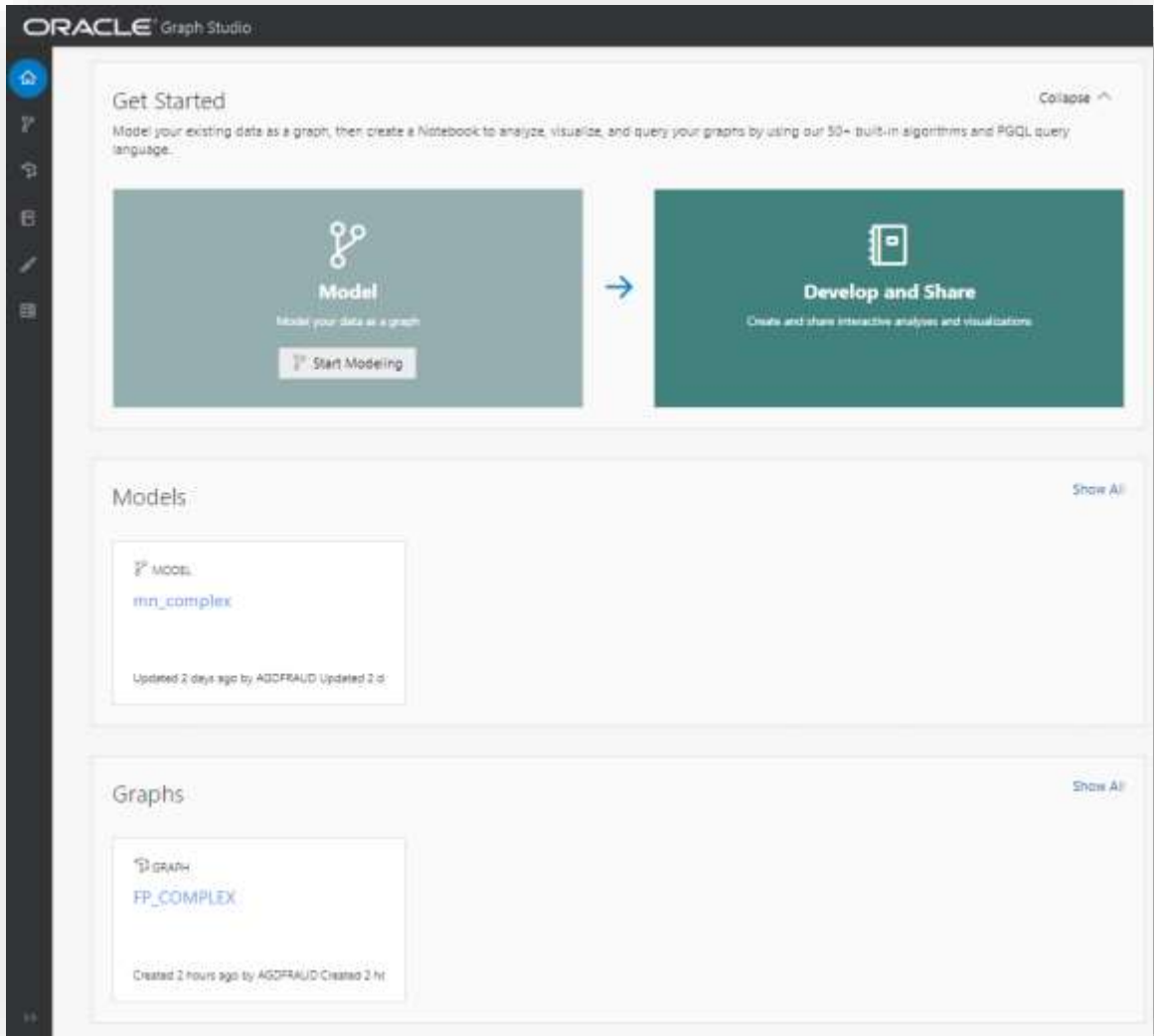
	A	B	C
1	acct_id	acct_owner_id	acct_name
2	10166	166	Ilyich Investment Holdings
3	98166	266	Dacha Retirement Planning
4	30366	366	Anti-Corruption Specialists Inc.
5	98366	366	Fraud Assurance LLC
6	40466	466	Bordeaux Oenoophilia GmbH
7	98466	466	JLR Legal Defense Fund
8	50566	566	Cotroceni Civic Holdings

Quickly, now:  
Prove that these  
transactions  
show distinct  
evidence of  
money  
laundering

	A	B
1	tt_type	tt_desc
2	CHR	Charitable Donation
3	CON	Consulting
4	CSX	Cash Transfer
5	LDF	Legal Defense Fund
6	REP	Real Estate Purchase
7	RES	Real Estate Sale
8	RFC	Retirement Fund Contribution
9	RFW	Retirement Fund Withdrawal
10	TRV	Travel Expenses

# Accessing Graph Studio Tools

- 1 You can **model** new graphs from tables and views, as well as **visualize** the materialized graphs ...

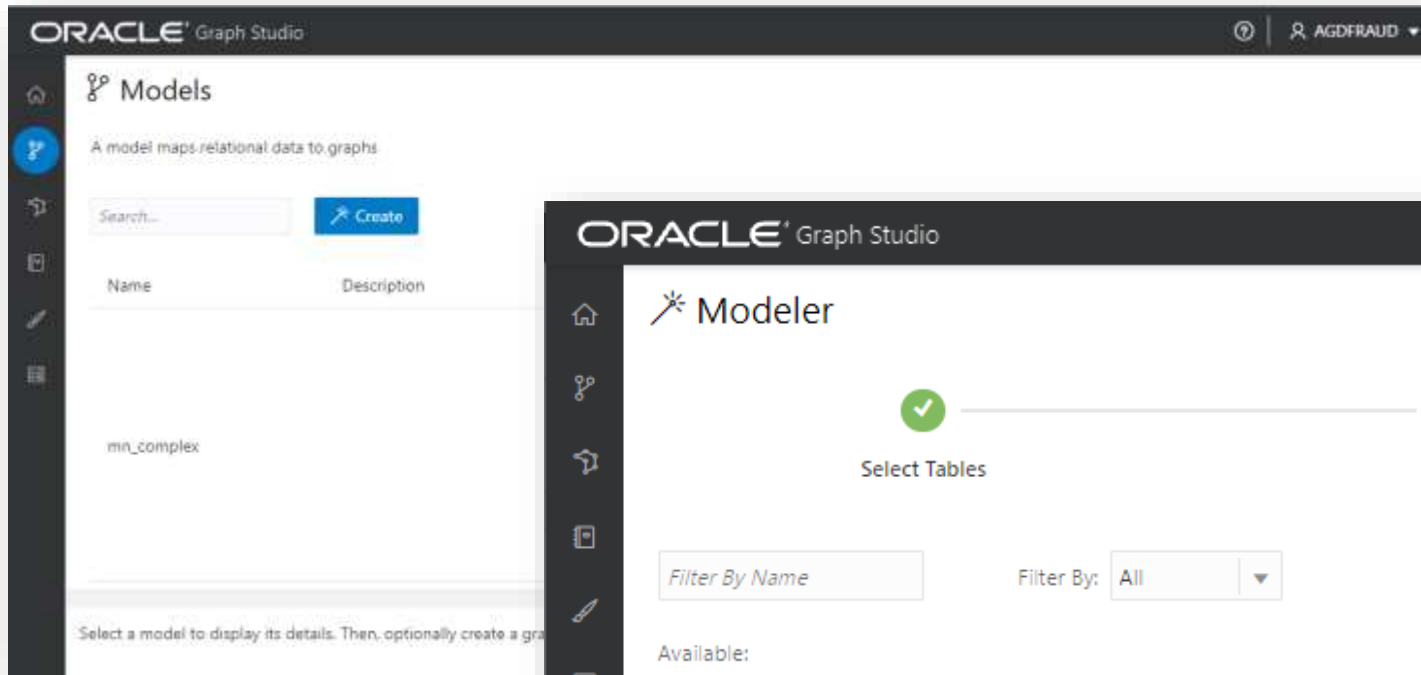


- 2 ... access existing Zeppelin **notebooks**, and review any **tasks** that have recently executed

# Creating and Accessing Property Graphs (1)

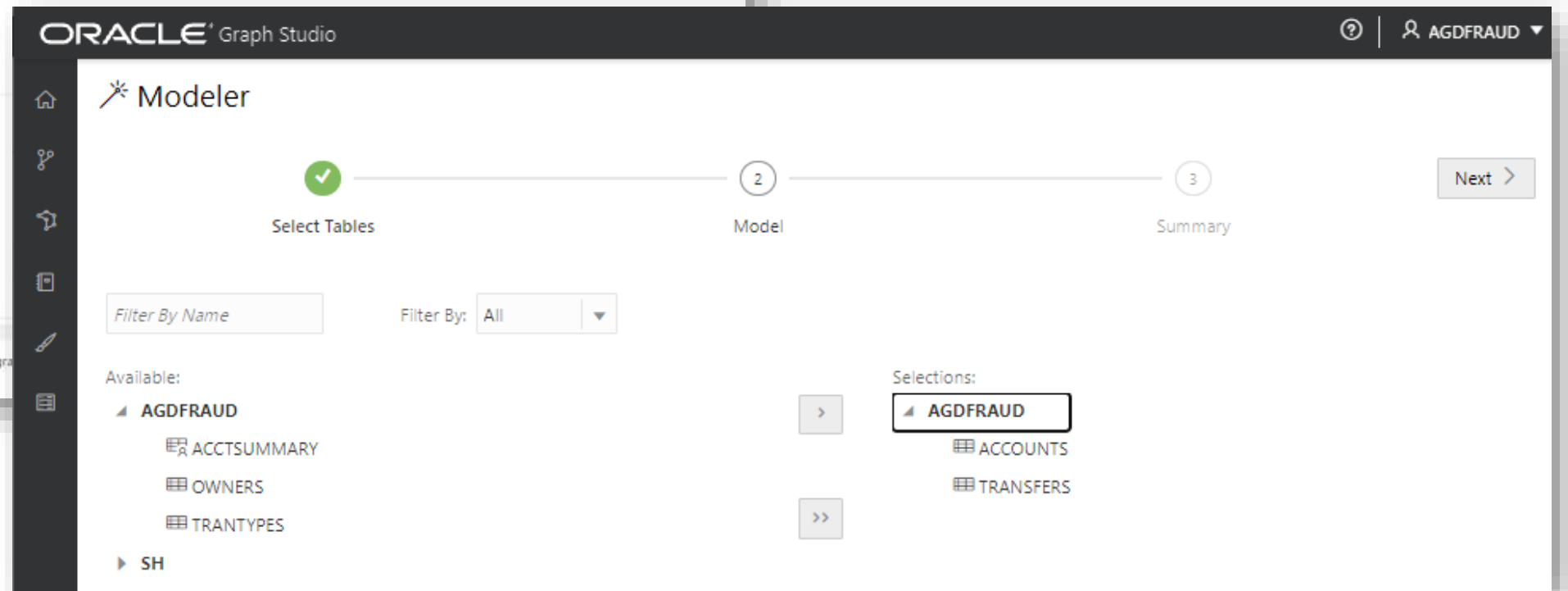
1

It's easy to model a new Property Graph from existing database tables and views



2

Once entities are chosen, Oracle automatically builds the **CREATE PROPERTY GRAPH** PGQL commands to create the graph using existing PK and FK constraints



# Creating and Accessing Property Graphs (2)

3 You can display properties from each **Vertex** ...

The screenshot shows the Oracle Graph Studio interface. The top navigation bar includes a home icon, a question mark, and the user name 'AGDFRAUD'. The main header is 'Modeler'. Below it is a progress bar with three steps: 'Select Tables' (checked), 'Model' (checked), and 'Summary' (3). The 'Designer' tab is active. On the left, 'Input Tables / Views (2)' lists 'AGDFRAUD.ACCOUNTS' and 'AGDFRAUD.TRANSFERS'. In the center, 'Vertex Tables (1)' has 'ACCOUNTS' selected. Below it, 'Edge Tables (1)' has a button '→ TRANSFERS'. At the bottom left, 'Source Table' is 'AGDFRAUD.ACCOUNTS' and 'Vertex Key' is 'ACCT\_ID'. On the right, 'Vertex Label' is 'ACCOUNTS' and 'Vertex Properties (3)' are listed in a table:

Name	Source Column
ACCT_ID	# ACCT_ID
ACCT_NAME	A ACCT_NAME
ACCT_OWNER_ID	# ACCT_OWNER_ID

4 ... as well as each **Edge** in the Property Graph, and even **exclude** specific columns from the final Graph

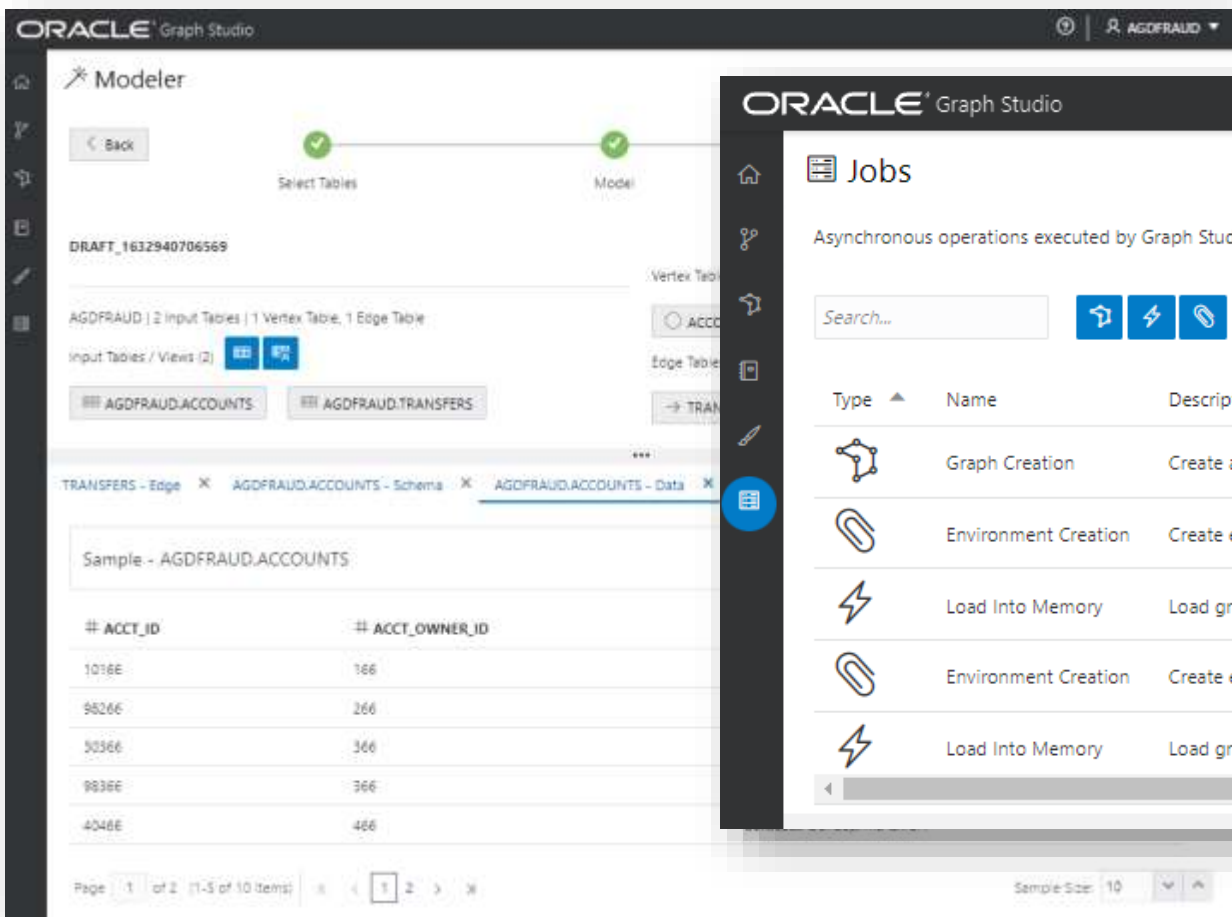
The screenshot shows the Oracle Graph Studio interface. The top navigation bar includes a home icon, a question mark, and the user name 'AGDFRAUD'. The main header is 'Modeler'. Below it is a progress bar with three steps: 'Select Tables' (checked), 'Model' (checked), and 'Summary' (3). The 'Designer' tab is active. On the left, 'Input Tables / Views (2)' lists 'AGDFRAUD.ACCOUNTS' and 'AGDFRAUD.TRANSFERS'. In the center, 'Vertex Tables (1)' has 'ACCOUNTS' selected. Below it, 'Edge Tables (1)' has a button '→ TRANSFERS'. At the bottom left, 'Source Table' is 'AGDFRAUD.TRANSFERS'. On the right, 'Edge Label' is 'TRANSFERS' and 'Edge Properties (6)' are listed in a table:

Name	Source Column	Options
SRC_ACCT_ID	# SRC_ACCT_ID	
TGT_ACCT_ID	# TGT_ACCT_ID	
TX_AMOUNT	# TX_AMOUNT	

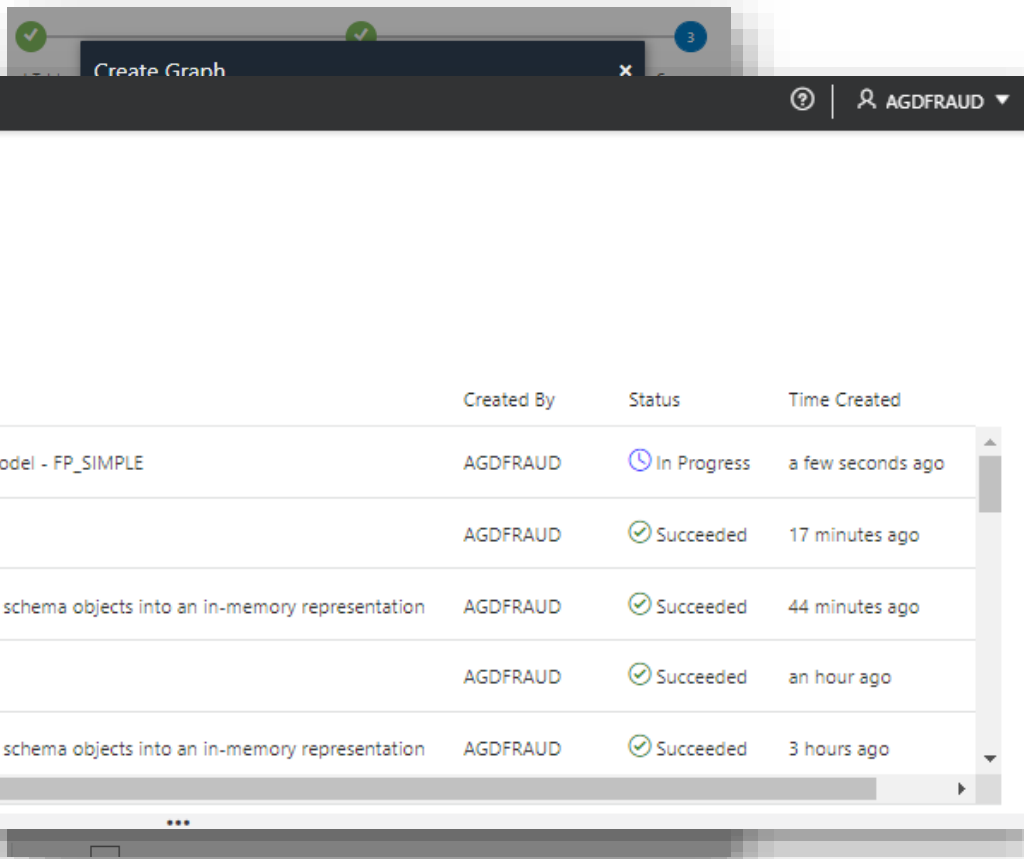


# Creating and Accessing Property Graphs (3)

5 Glimpse the **data** within each **Vertex** and **Edge** ...



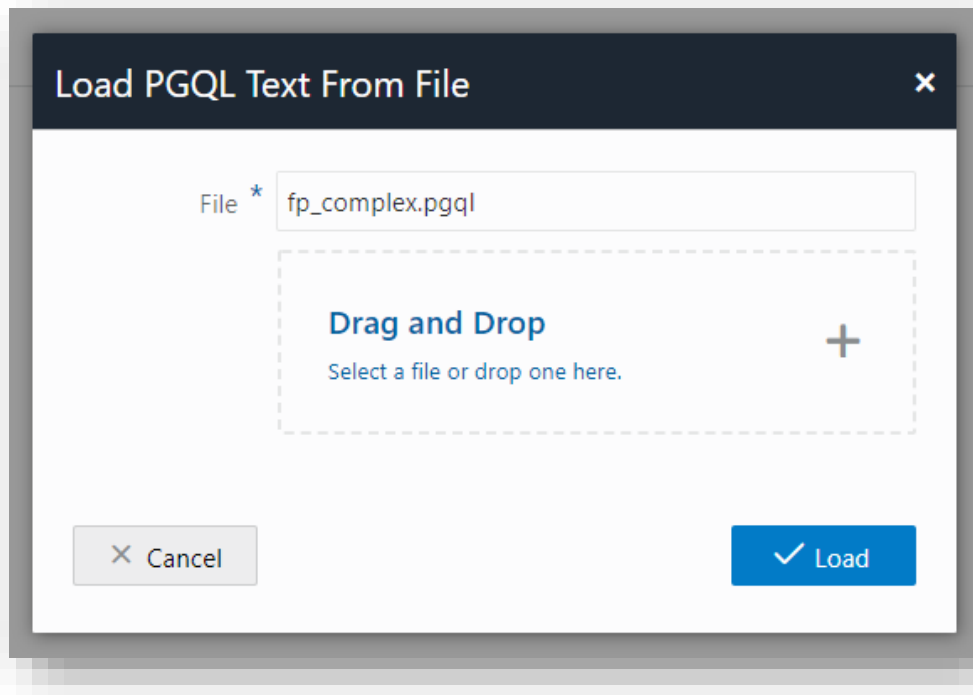
6 ... supply **names** and **descriptors** for the new Model and Graph ...



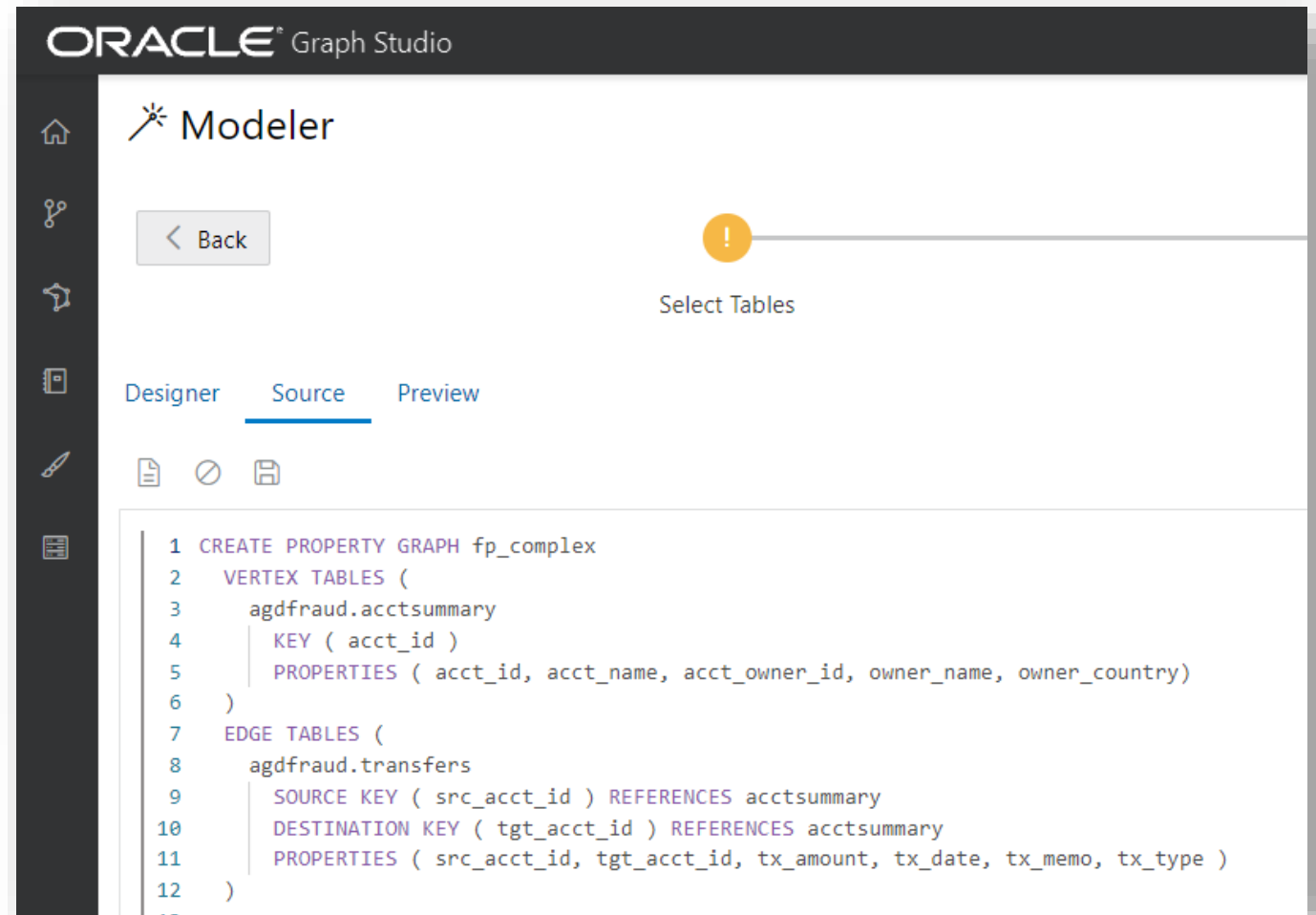
7 ... and **monitor** the successful creation of the new Graph and its corresponding Model

## Creating and Accessing Property Graphs (4)

- 8 If you have a PGQL file containing a **CREATE PROPERTY GRAPH** statement ...



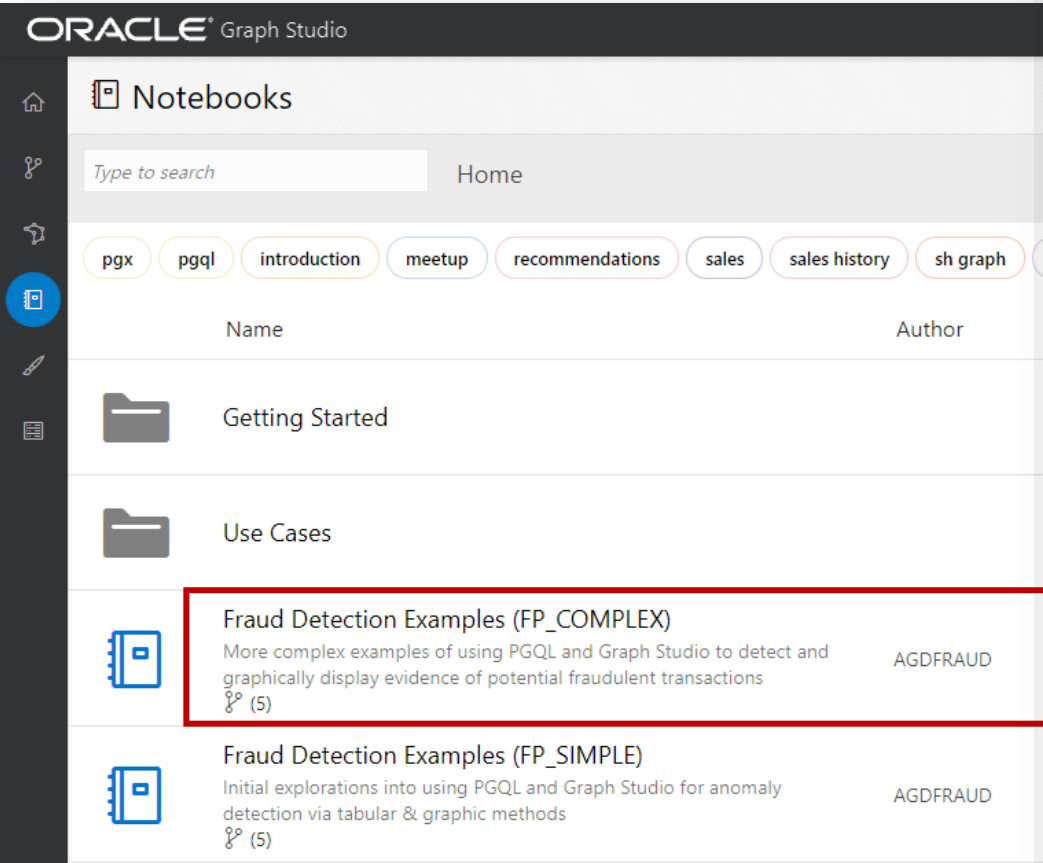
- 9 ... you can supply it within the Modeler instead of building the Graph and Model graphically



# Leveraging Zeppelin Notebooks To Probe & Display Property Graphs (1)

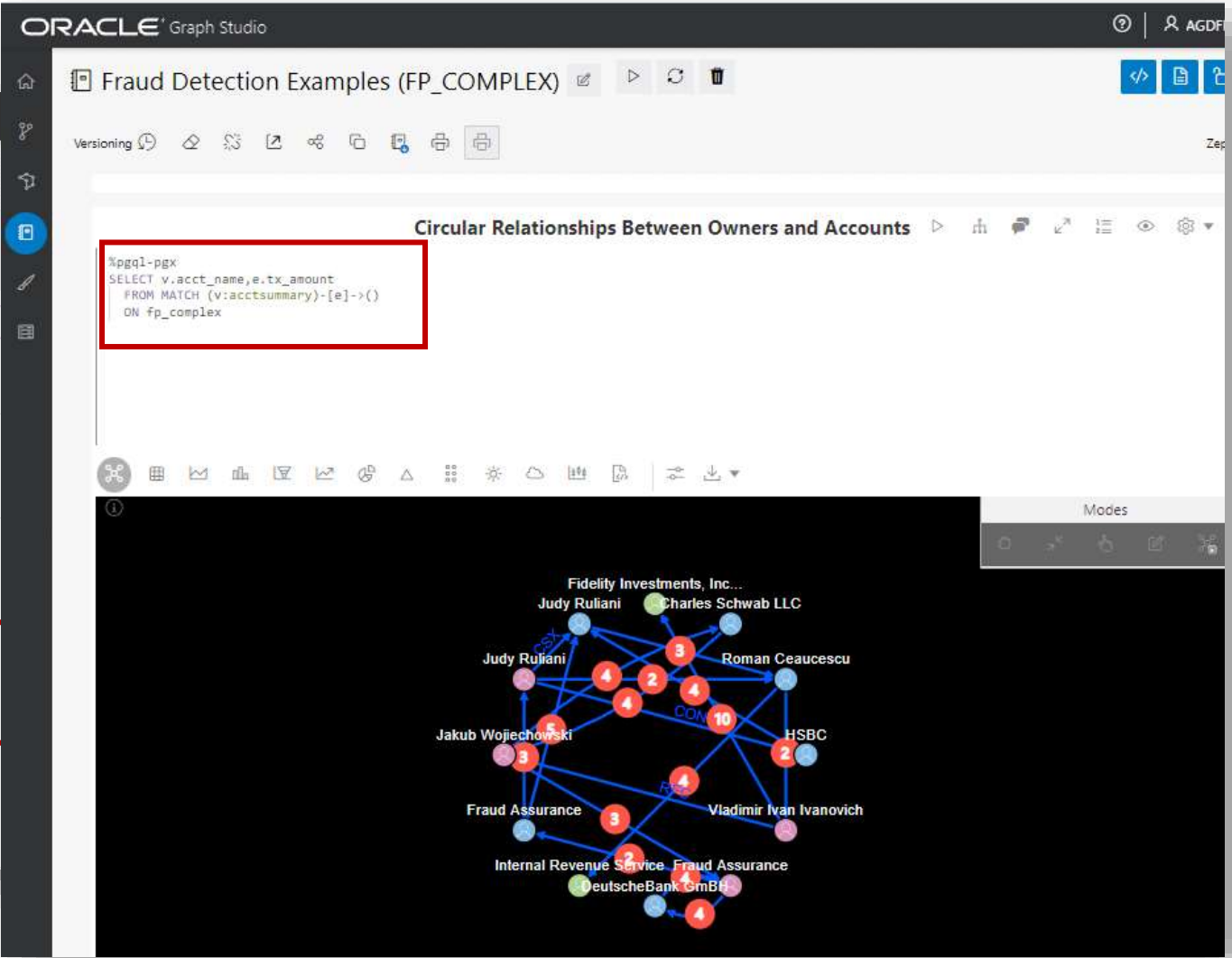
1

Zeppelin notebook technology allows even inexperienced PGQL users to immediately dive into property graph content ...



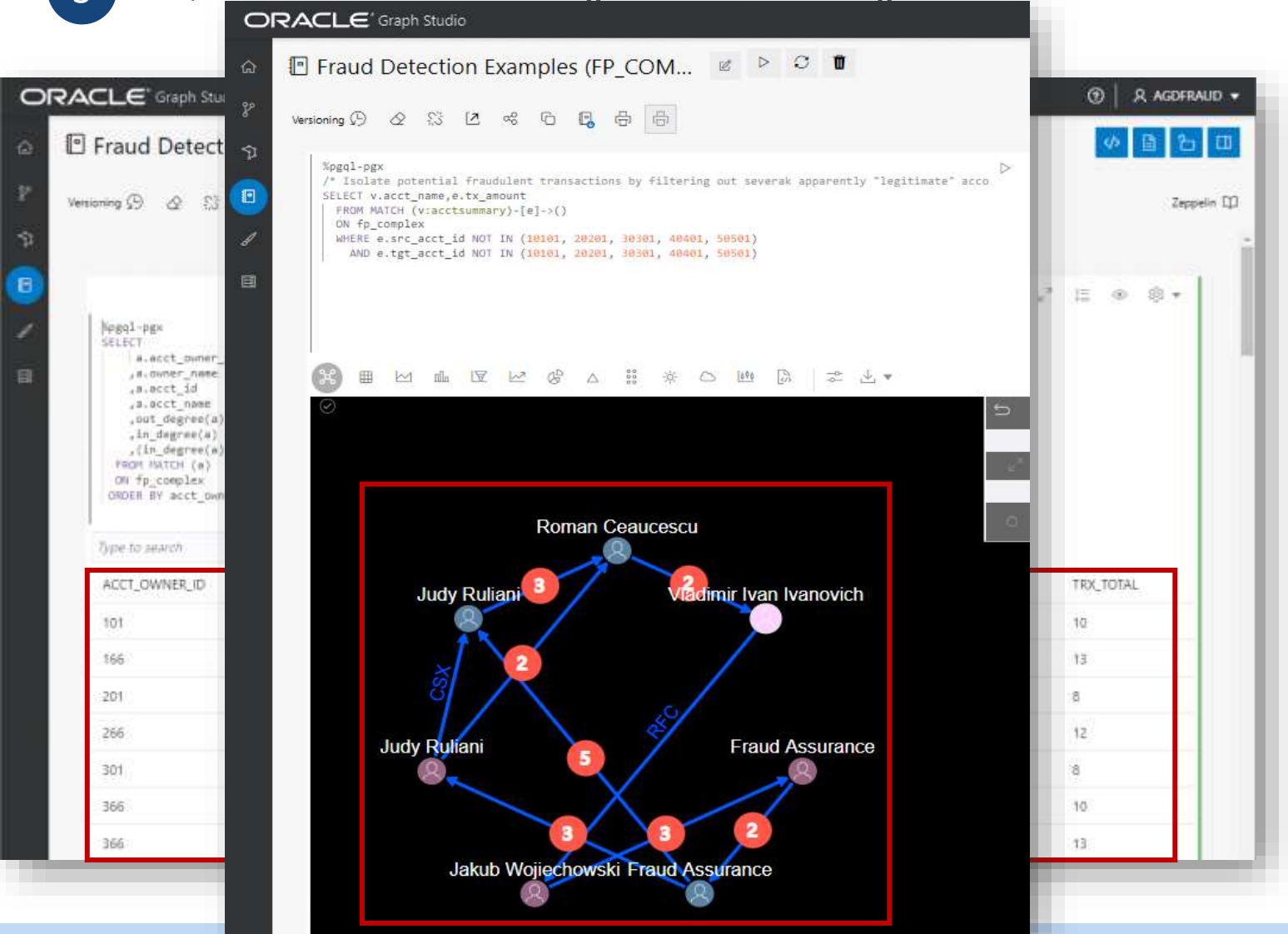
2

... and with a simple PGQL statement and some mouse clicks, suspicious patterns are immediately evident!



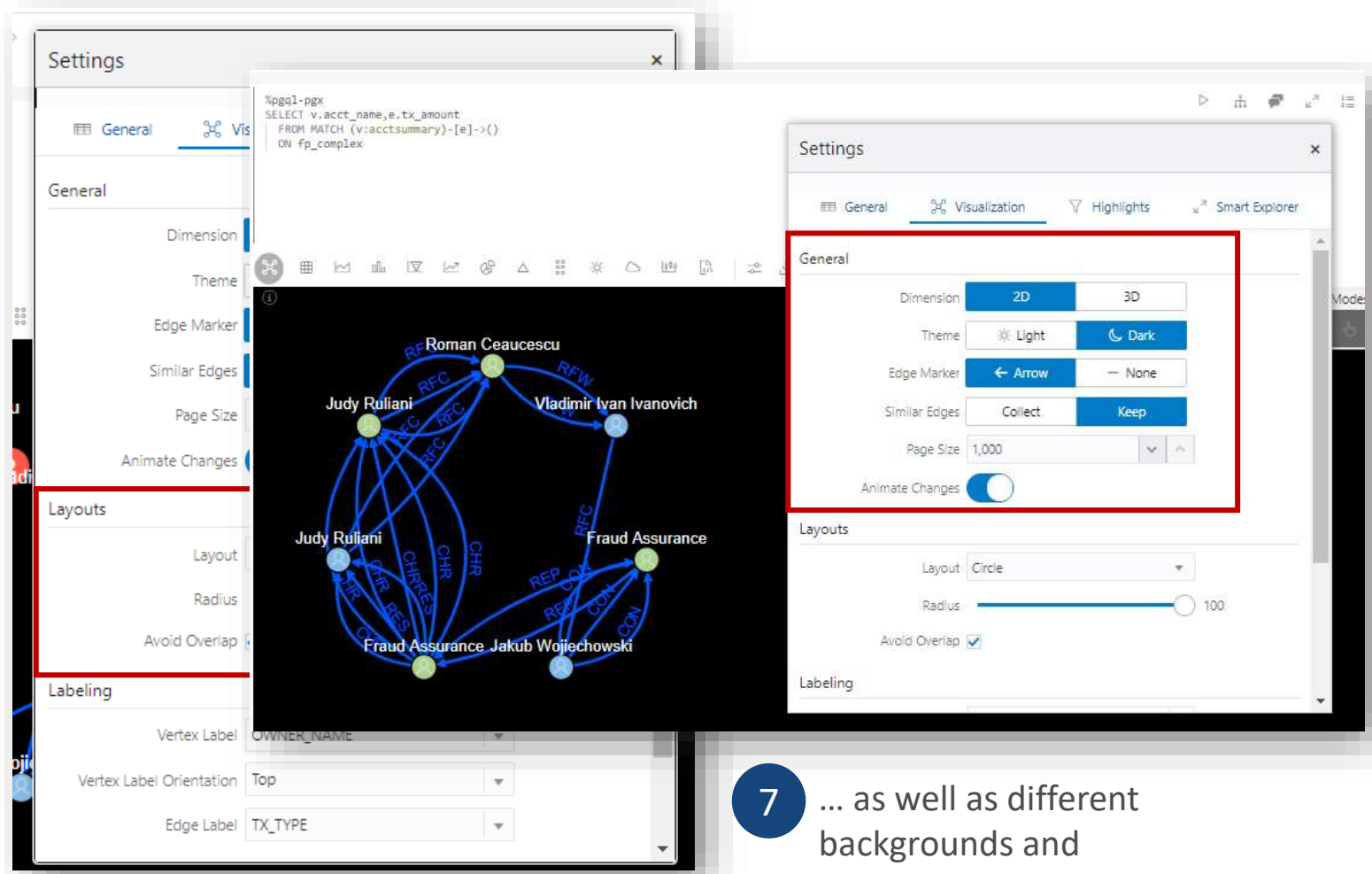
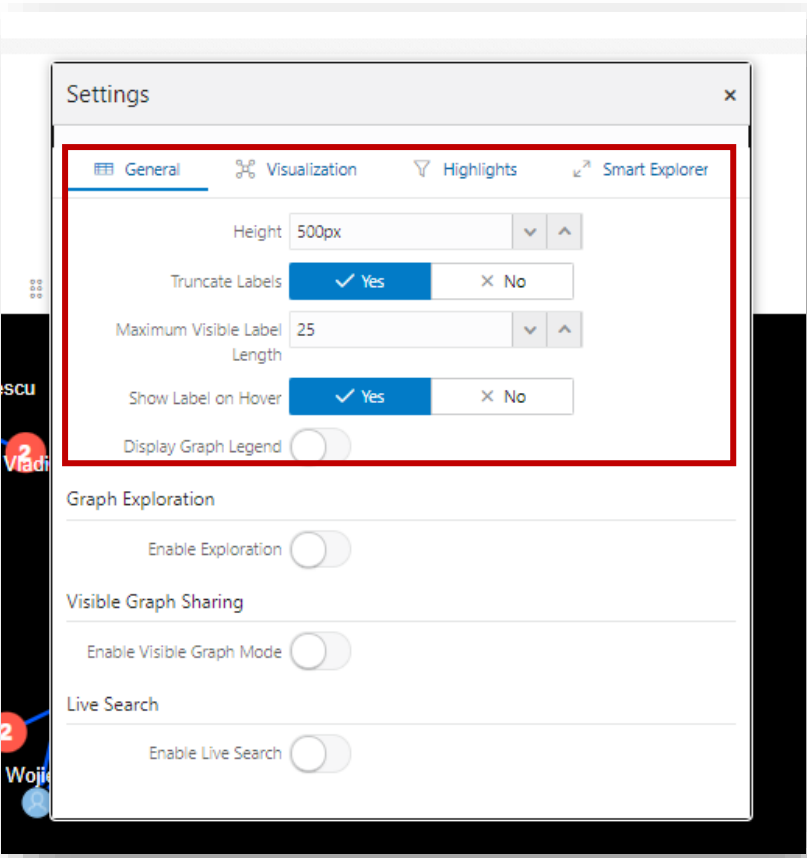
# Leveraging Zeppelin Notebooks To Probe & Display Property Graphs (2)

4 ... and, with just a little polish the currently fraudulent activity becomes even more evident..



# Leveraging Zeppelin Notebooks To Probe & Display Property Graphs (3)

5 Property graph display options are flexible and precise ...



7 ... as well as different backgrounds and representations of vertices



# Live Demonstration: *Like, Wow, That's Amazing, Miss Information!*



## Just 12 People Are Behind Most Vaccine Hoaxes On Social Media, Research Shows

Updated May 14, 2021 - 11:48 AM ET

Heard on All Things Considered



SHANNON BOND



To illustrate how easy it is to leverage **Graph Studio**, here's an example of how to **detect patterns** in (fictitious!) social media postings and possibly identify which Twitter accounts **are being spread by bots and "sock puppets"** versus posts by **actual human beings**



What could  
possibly go wrong?



# Beyond PGQL: Other Property Graph Tools

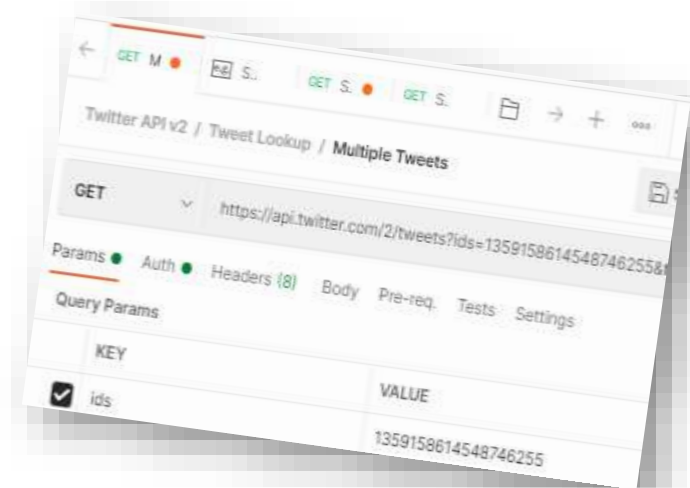
Property Graph toolsets can be accessed through just about any programming language, including **Java**, **Python**, and even **SQLcl** and **PL/SQL**



**60+** powerful **graph algorithms** offer the ability to:

- Detect graph **components** and **communities**
- Evaluate graph **structures** for patterns
- **Rank** and “**walk**” graphs
- **Identify paths** through graph nodes
- Build **machine learning models**

Leverage the **Graph Client API** to build custom applications



## PGX Analytic Functions Provide a Deeper Look *Within* Graphs

Analytic Function	Provides Useful Intelligence For:
PageRank	Determines <i>which node in a graph is most important</i> based on its number of incoming edges
Closeness Centrality	Calculates <i>how “close”</i> a node is to other nodes within a graph
Betweenness Centrality	Detects <i>how much a node may influence</i> the flow of information within a graph
HITS	Points to which <i>web page</i> is <i>likely to contain the most meaningful information</i> based on its <i>HyperLink-Induced Topic Search</i> score
WTF	Projects <i>Whom To Follow</i> within a social network for <i>maximum positive impact to social standing</i>

## Using PGX To Enhance Graph Content

**1** You can apply any one of **60+** additional graph **analytic functions** to an **existing** in-memory graph

2 ... and you can use these new attributes for further **analyses** or even different **visualizations**

Handle	# of Followers	# of Following	PageRank	DistBtwn	Hits - Authority	Hits - Hubs
@Eagle1001	20	5	0.0034418799889605984	118.0	0.027406761249333042	0.4854152337546281
@Patriot987	50	35	0.0030163043478260874	79.0	5.4171665766392065E-188	0.1545814580986151
@Patriot1007	21	18	0.0022713994565217396	74.0	0.01774315361502717	0.860344755549369
@AntiAntifa1009	52	12	0.0027540718410326093	44.000000000000001	0.5115031862658065	0.01683820803240701
@Deplorable448	6	80	0.00172696925951087	27.0	0.09875200653015585	3.204579685542998E-141
@AntiAntifa484	9	22	0.00172696925951087	27.0	0.09875200653015585	3.204579685542998E-141

# PGX ML Toolset (And You Thought “Normal” ML Was Tough To Grok!)

Model	Description	Examples of Real-World Use Cases
<u><a href="#">DeepWalk</a></u>	Computes <b>random walks</b> for every vertex, then generates <b>new</b> vector representations	How likely is it that a <i>new post</i> will spread quickly through a <b>network of connected friends</b> via a mobile social media application?
<u><a href="#">Supervised GraphWise</a></u>	Based on <b>GraphSage</b> , it's an inductive vertex representation learning algorithm against <b>vertex feature information</b>	Based on a customer's <i>prior ordering habits</i> , what new products or offerings can we suggest <b>that they're actually interested in</b> ?
<u><a href="#">Unsupervised GraphWise</a></u>	Based on <b>Deep Graph Infomax</b> , it applies an inductive vertex representation learning algorithm against vertex information	Can we quickly <b>identify brain abnormalities to detect autism spectrum disorder</b> (ASD) by comparing 4-D MRI brain scans of new patients against those of patients <i>already diagnosed with ASD</i> ?
<u><a href="#">Pg2Vec</a></u>	Generates <b>graphlets</b> that can be compared for <b>matching patterns</b>	Based on <i>prior known patterns</i> , is a new set of financial transactions a warning sign that <b>money laundering may be occurring</b> ?



## Plans for Future Experimentation



**Expand beyond** Graph Studio for Autonomous Database to use powerful PGX tools **in native mode**



Use publicly-available **social media data** to refine methods **identifying spread of misinformation**



Leverage data **captured from Twitter in real time** to explore available **Machine Learning** algorithms

# Sample Use Cases For Property Graphs

- **Social Media Sentiment Analysis**

<https://towardsdatascience.com/sentiment-analysis-74624b075842>

- **Graphs Analytics for Fraud Detection**

<https://towardsdatascience.com/graphs-analytics-for-fraud-detection-83ee3af81ec7>

- **Detecting Fake Users on Social Media with a Graph Database**

<https://journals.uvic.ca/index.php/arbutus/article/view/20027>

- **Just 12 People Are Behind Most Vaccine Hoaxes On Social Media, Research Shows**

<https://www.npr.org/2021/05/13/996570855/disinformation-dozen-test-facebooks-twitters-ability-to-curb-vaccine-hoaxes>

# Useful References

- **Graph Databases and Analytics: How to Use Them**

<https://www.oracle.com/it/a/tech/docs/sg-oow2019-graph-databases-and-analytics.pdf>

- **Property Graph Developer's Guide**

<https://docs.oracle.com/en/database/oracle/property-graph/20.4/spgdg/oracle-graph-property-graph-developers-guide.pdf>

- **PGQL: Vertex and Edge Functions**

<https://pgql-lang.org/spec/1.4/#vertex-and-edge-functions>

- **Using the Machine Learning Library (Pgxml) for Graphs**

<https://docs.oracle.com/en/database/oracle/property-graph/22.1/spgdg/using-machine-learning-library-pgxml-graphs.html>