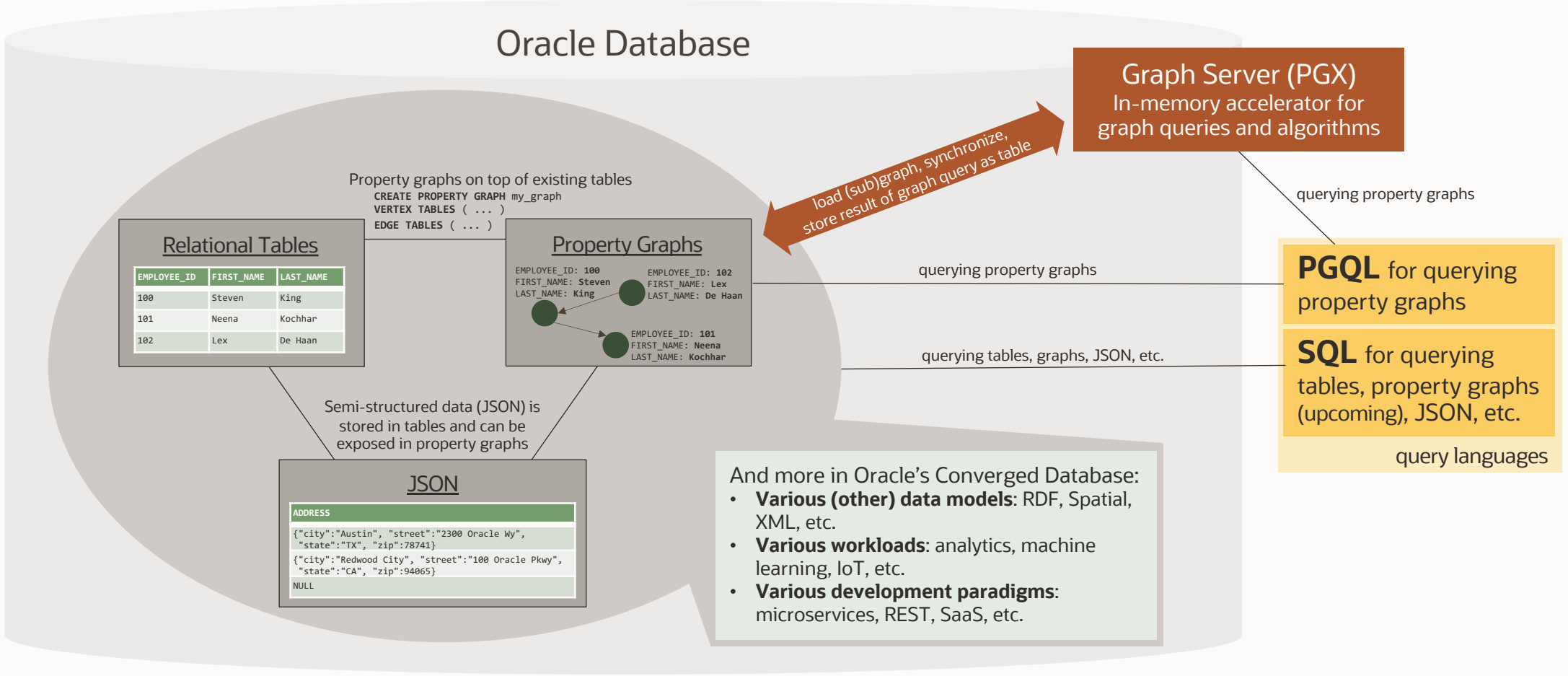
ORACLE

Creating and Querying Property Graphs in Oracle, On-Premise and in the Cloud

Fifteenth LDBC TUC Meeting

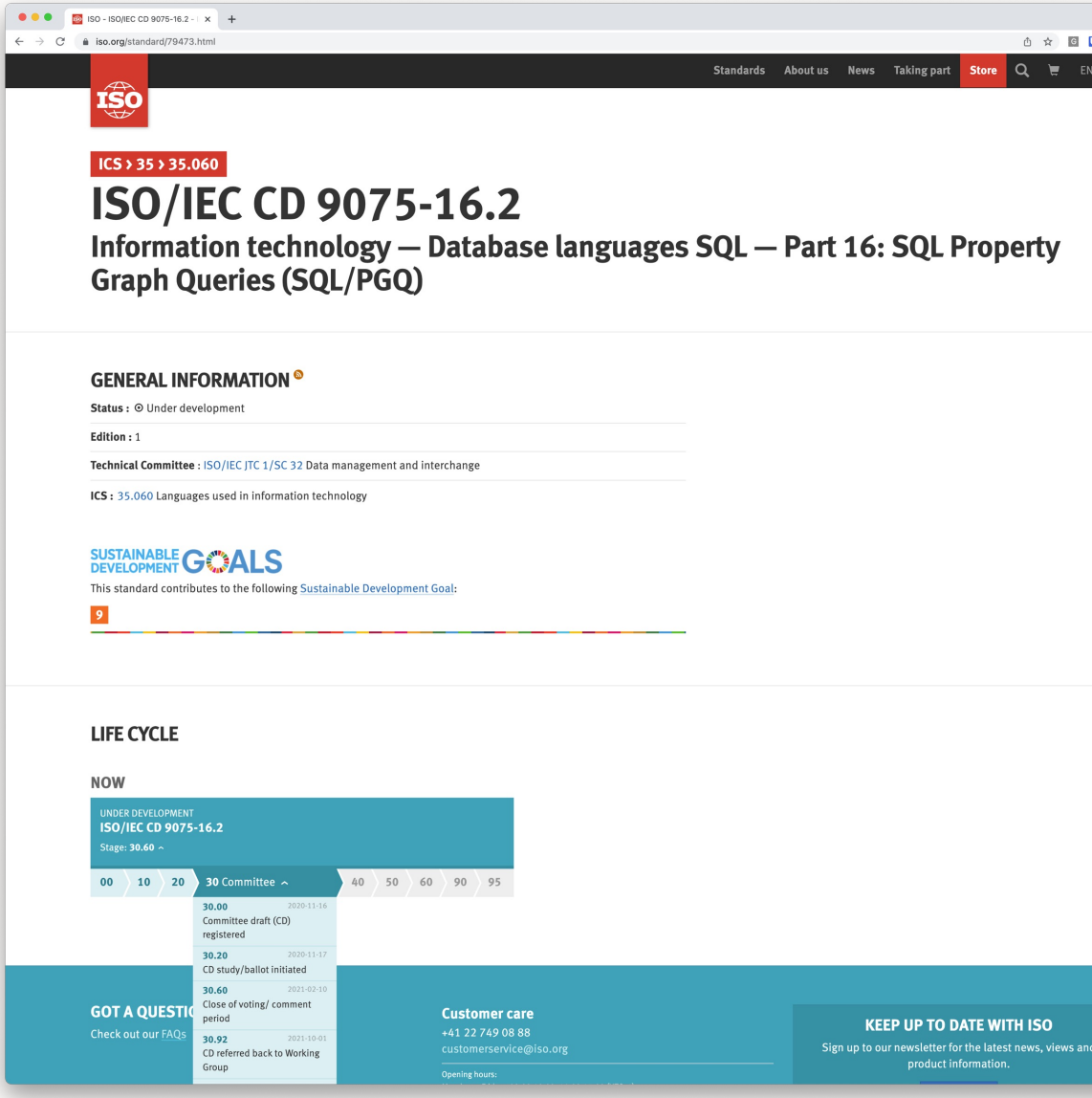
Oskar van Rest
Consulting Member of Technical Staff
Oracle Property Graph Technology
June 17, 2022

Property Graphs in Oracle's Converged Database



Upcoming Property Graph extension for the SQL standard

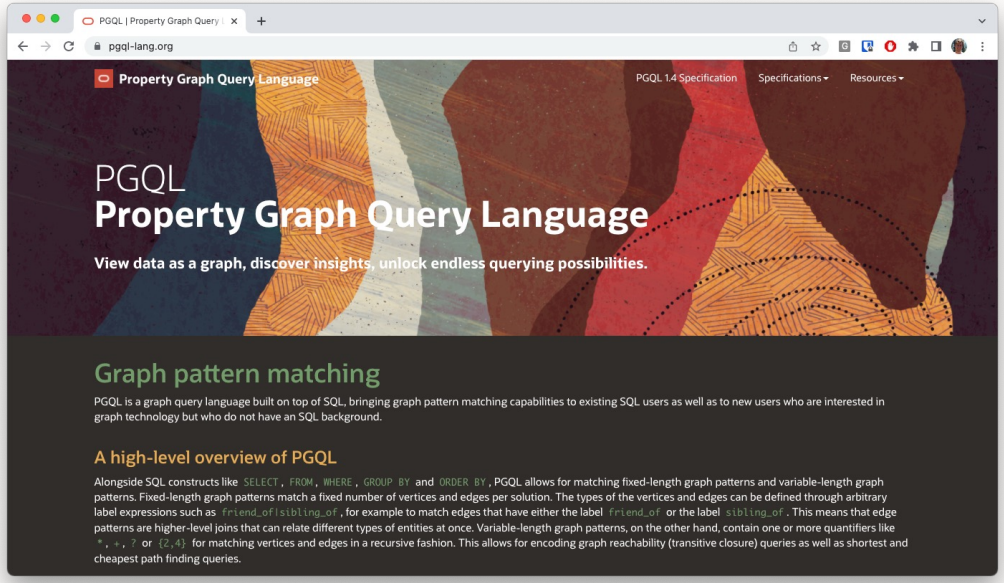
- SQL is one of the two languages being developed by the ISO/IEC JTC 1/SC 32 “Data management and interchange” committee (GQL being the other one)
- SQL/PGQ is the Property Graph extension for SQL
- SQL/PGQ release timeline:
  - Major functionality finalized by June 2022
  - Only bug fixes are accepted afterwards
  - ISO publication expected in spring 2023
- Functionality included in first release:
  - Capability to create graphs on top of existing relational tables/views (CREATE PROPERTY GRAPH)
  - Capability to query graphs
    - Fixed-length pattern matching (MATCH)
    - Variable-length pattern matching (ANY, SHORTEST, ...)



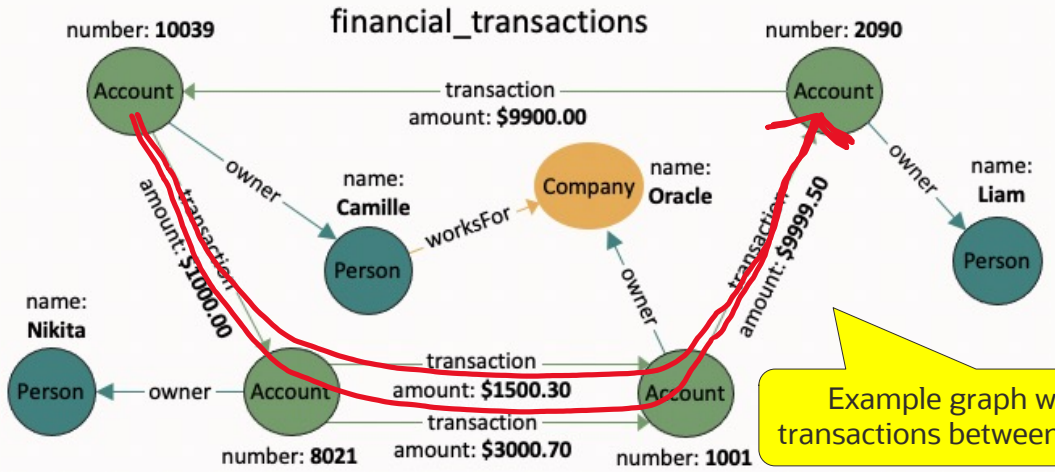
https://www.iso.org/standard/79473.html

Property Graph Query Language (PGQL)

- PGQL is our current approach to querying property graphs in Oracle Property Graph
- PGQL is further developed in alignment to SQL/PGQ
- Latest extensions to PGQL:
  - CREATE PROPERTY GRAPH and DROP PROPERTY GRAPH statements
  - Unnesting of variable-length paths
  - Path-finding goals ANY, ALL, ALL SHORTEST
  - SELECT x.* for selecting all properties of a vertex/edge
  - String functions LISTAGG, concat (||), UPPER, LOWER, SUBSTRING



https://pgql-lang.org/



Example graph with financial transactions between bank accounts

```
SELECT CAST(a.number AS STRING) || ', ' || LISTAGG(x.number, ', ')
         AS account_numbers_along_path
FROM MATCH ALL SHORTEST (a:Account)
                          (-[e:transaction]-> (x:Account))*
                            (b:Account)
       ON financial_transactions
WHERE a.number = 10039 AND b.number = 2090
ORDER BY account_numbers_along_path
```

Example query: "Find all shortest paths between bank accounts 10039 and 2090. For each path, show the account numbers."

```
+----------------------------+
| account_numbers_along_path |
+----------------------------+
| 10039, 8021, 1001, 2090    |
| 10039, 8021, 1001, 2090    |
+----------------------------+
```

Output for example query, showing the bank account numbers along both paths that were matched

CREATE PROPERTY GRAPH statement (1/2)

Example: create a property graph from tables based on predefined primary and foreign keys

```
CREATE PROPERTY GRAPH student_network
  VERTEX TABLES ( Person PROPERTIES ( name, dob ),
                  University PROPERTIES ( name ) )
  EDGE TABLES ( knows SOURCE KEY ( person1_id ) REFERENCES Person ( id )
                      DESTINATION KEY ( person2_id ) REFERENCES Person ( id )
                      NO PROPERTIES,
                studentOf SOURCE Person
                          DESTINATION University
                          NO PROPERTIES )
```

Vertex tables:

Person

id	name	dob
1	Riya	1995-03-20
2	Kathrine	1994-01-15
3	Lee	1996-01-29

University

id	name
1	UC Berkeley

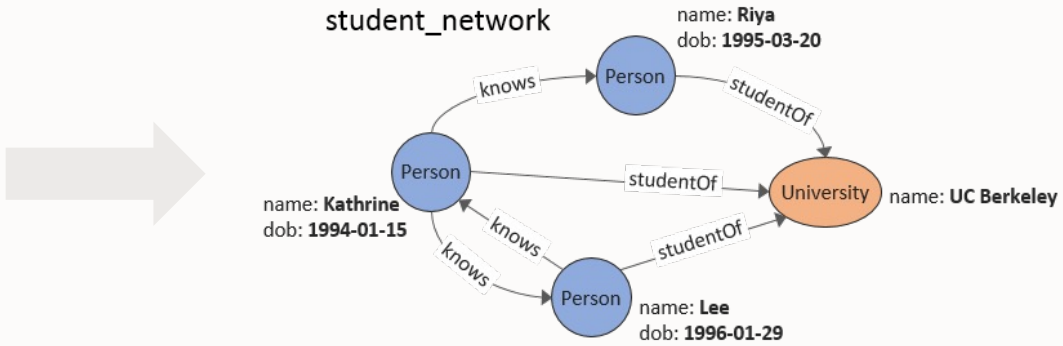
Edge tables:

knows

person1_id	person2_id
2	1
2	3
3	2

studentOf

person_id	university_id
1	1
2	1
3	1



CREATE PROPERTY GRAPH statement (2/2)

Example: create a property graph from tables while manually specifying keys for vertices and edges (needed when e.g., primary and foreign keys are not defined)

```
CREATE PROPERTY GRAPH student_network
  VERTEX TABLES ( Person KEY ( id ) PROPERTIES ( name, dob ),
                  University KEY ( id ) PROPERTIES ( name ) )
  EDGE TABLES ( knows KEY ( person1_id, person2_id )
                      SOURCE KEY ( person1_id ) REFERENCES Person ( id )
                      DESTINATION KEY ( person2_id ) REFERENCES Person ( id )
                      NO PROPERTIES,
                studentOf KEY ( person_id, university_id )
                          SOURCE KEY ( person_id ) REFERENCES Person ( id )
                          DESTINATION KEY ( university_id ) REFERENCES University ( id )
                          NO PROPERTIES )
```

Vertex tables:

Person

id	name	dob
1	Riya	1995-03-20
2	Kathrine	1994-01-15
3	Lee	1996-01-29

University

id	name
1	UC Berkeley

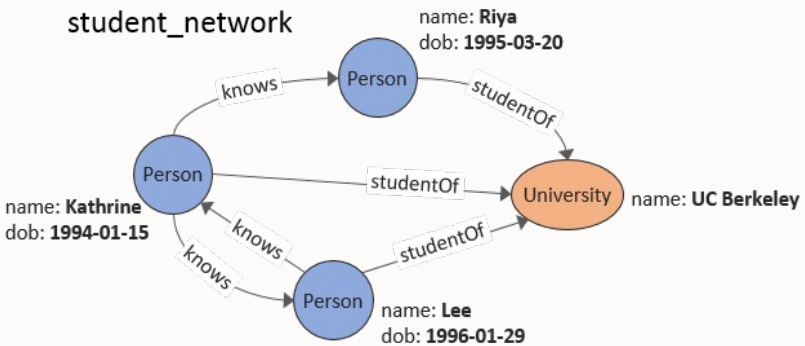
Edge tables:

knows

person1_id	person2_id
2	1
2	3
3	2

studentOf

person_id	university_id
1	1
2	1
3	1



Property graphs + JSON

```
ERROR: oracle.pgx.api.MalformedQueryException: Error(s) in line 1:

SELECT n.firstName FROM MATCH (n:Post|Comment)
         ^^^^^^^^^
Property does not exist for any of the labels
```

- Tabular property graphs are statically typed
  - Provides helpful errors when a user misspells a vertex/edge label or property name
  - Easier to optimize performance
  - But: some use cases have semi-structured data
- JSON is our recommended approach to managing semi-structured data in property graphs
  - Oracle database release 21c introduced a JSON datatype, which is an optimized native binary storage format[1]
    - Same functionality as for JSON strings (since Oracle Database 12c Release 1), but native data type give better performance
  - Example 1: JSON expressions in PGQL SELECT queries[2]

```
SELECT JSON_VALUE(n.address, '$.city') AS city
FROM MATCH (n:Person) WHERE n.id = 100
```

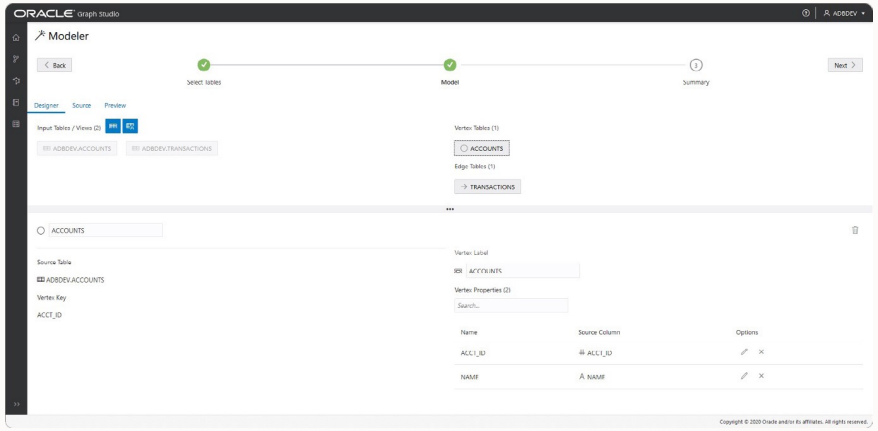
  - Example 2: field inside JSON document exposed as vertex/edge property (upcoming):

```
CREATE PROPERTY GRAPH my_graph
... PROPERTIES ( first_name, last_name, JSON_VALUE(n.address, '$.city') AS city )
```

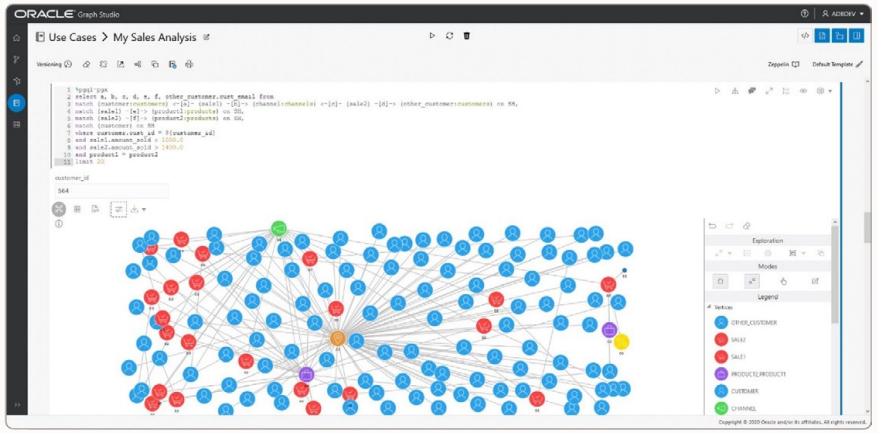
```
SELECT n.city
FROM MATCH (n:Person) WHERE n.id = 100
```

1. JSON DataType Support in Oracle 21c https://blogs.oracle.com/database/post/json-datatype-support-in-oracle-21c
2. Using JSON to Store Vertex and Edge Properties https://docs.oracle.com/en/database/oracle/property-graph/22.2/spgdg/using-json-store-vertex-and-edge-properties.html

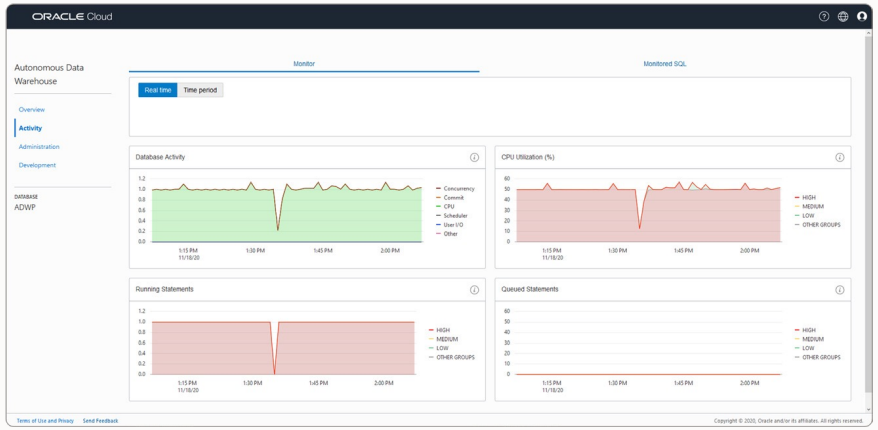
Property Graphs in the Oracle Cloud



- The Graph Modeler is a UI for creating graphs from tables.
- It generates a CREATE PROPERTY GRAPH statement that can be customized and executed.



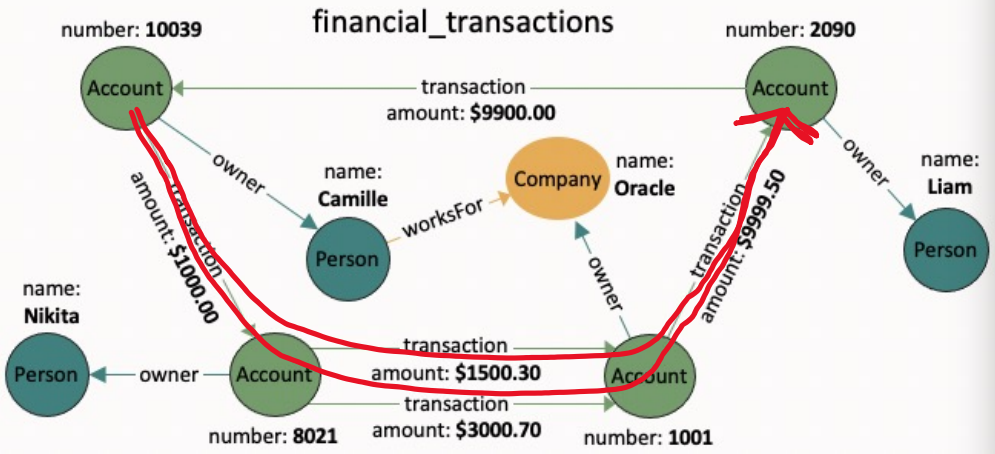
- Users can create notebooks with PGQL paragraphs.
- The result of a PGQL SELECT query is presented as graph, table or chart (e.g., bar chart).

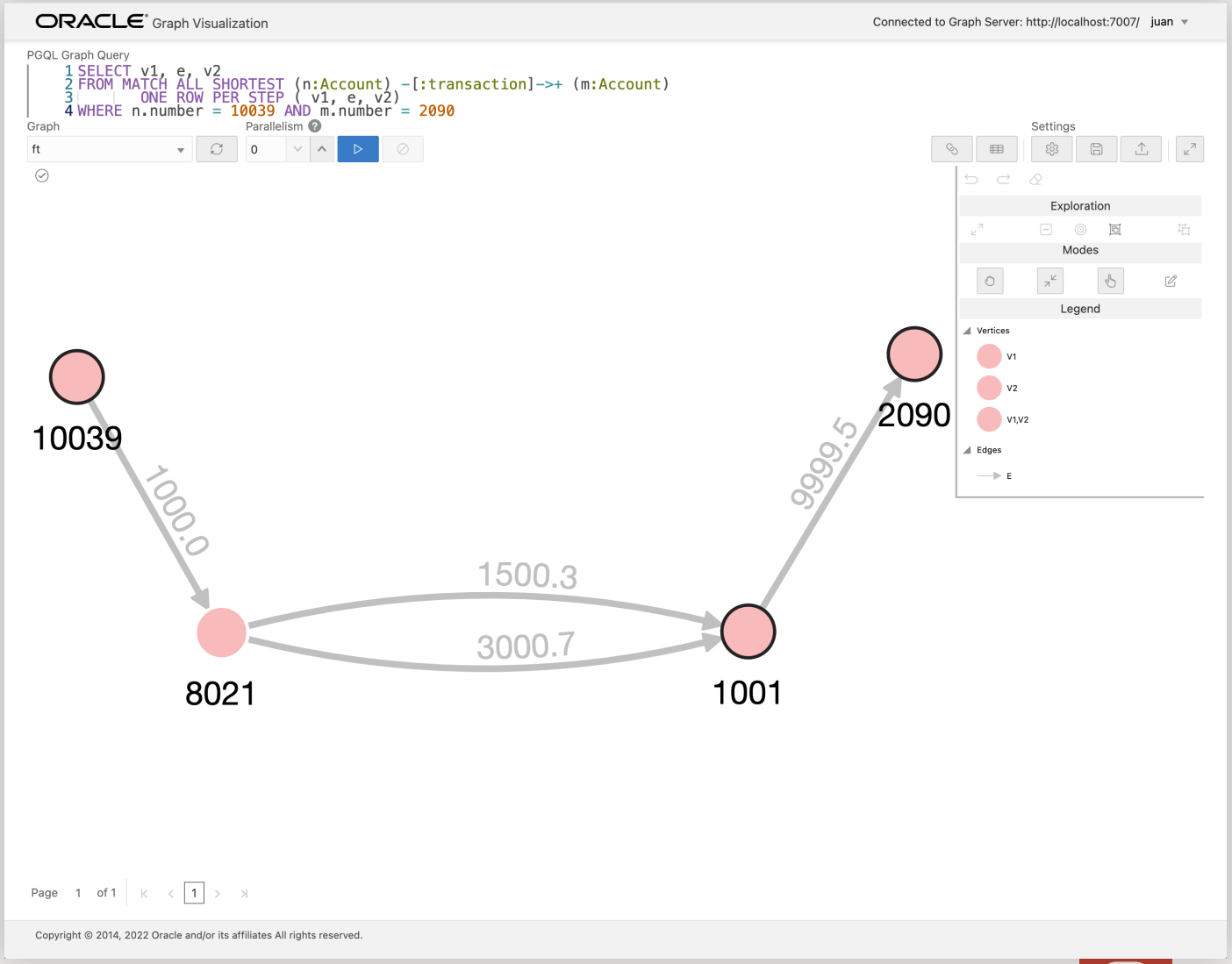


- Fully integrated into Oracle Autonomous Database

Oracle Graph Visualization

- Visualize a subgraph based on vertices and edges returned by a PGQL query
- Ability to expand neighbors
- Change icons, labels, colors
- Embed visualization as iframe on any website
- Etc.





Update on our experience with LDBC SNB queries

- Using LDBC to better understand the performance characteristics of different graph implementations at Oracle:
  - e.g. Oracle Graph Server (PGX) vs. translation into (PL/)SQL
- Can express all 20 BI queries (the most complex SNB workload) in PGQL
  - Several queries require “temporary vertex/edge properties” to store intermediate query results, via UPDATE queries. Ideally, we would not have to break up queries into multiple parts (need enhancements to PGQL).
- Ongoing PGQL work that will improve our LDBC implementation:
  - Subqueries in FROM clause (BI Q12, Q14)
    - Additionally, could benefit from a language construct like SQL’s common table expressions (BI Q4, Q13)
  - INTERVAL support
    - Currently working around by defining UDFs for adding days to a date (BI Q2) and hours to a timestamp (BI Q17)
- Continuing our work with SNB:
  - Want to use LDBC’s query driver and mix read and write queries
  - Want to cover both workloads (Interactive + Business Intelligence)

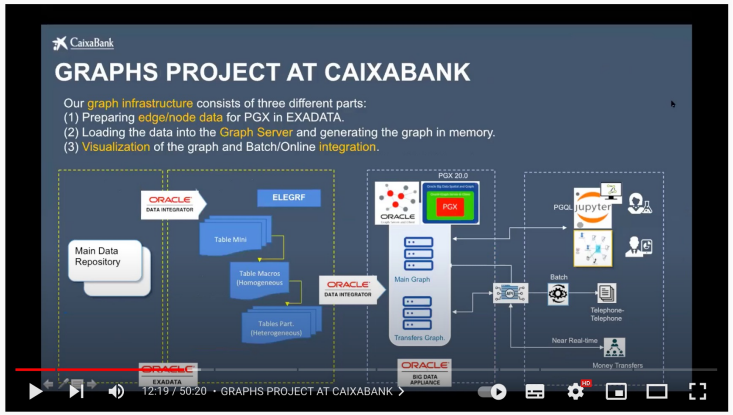
Success Stories from Customers and Partners

Paysafe:

https://youtu.be/rggYtCAeGUY

https://www.oracle.com/be/a/ocom/docs/paysafe-case-study.pdf

CaixaBank

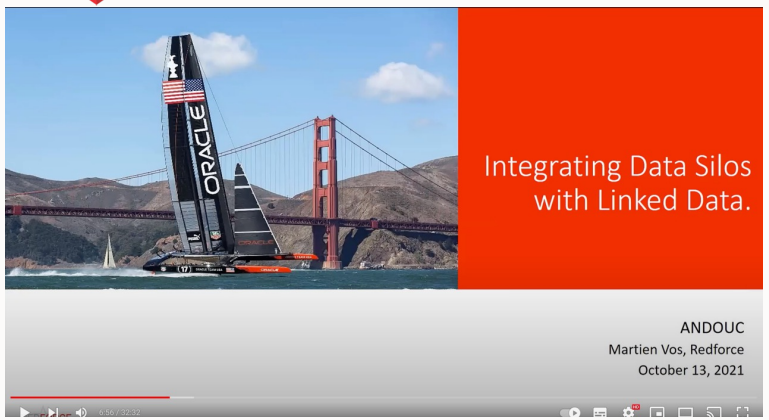


https://youtu.be/j_RlUmd6qps

PITSS
Simplifying Complexity

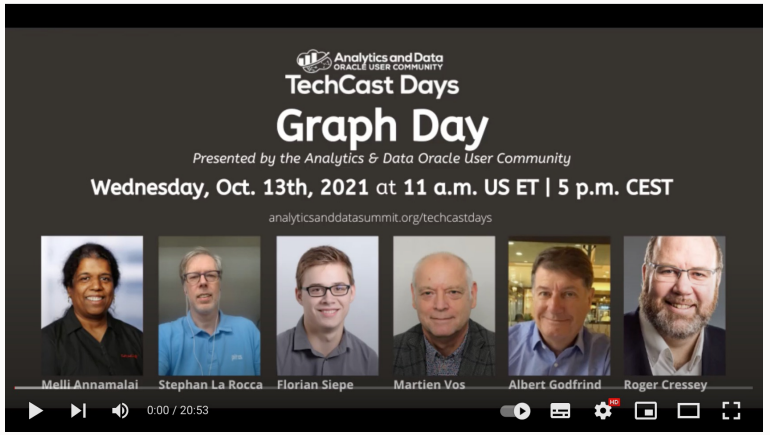
https://youtu.be/6pzXxvT8BRk

REDFORCE



https://youtu.be/w34U9Fnh3vQ

VIESMANN



https://youtu.be/trrmoUyOtwE

https://www.oracle.com/database/graph/

Thank you

OCI Services Solutions Why OCI Customers Pricing Learn Developers Support Marketplace Sign in to Oracle Cloud

Database ›

Try graph for free

Graph Database and Graph Analytics

Graph databases, part of Oracle's converged database offering, eliminate the need to set up a separate database and move data. Analysts and developers can perform fraud detection in banking, find connections and link to data, and improve traceability in smart manufacturing, all while gaining enterprise-grade security, ease of data ingestion, and strong support for data workloads.

Oracle Autonomous Database includes Graph Studio, with one-click provisioning, integrated tooling, and security. Graph Studio automates graph data management and simplifies modeling, analysis, and visualization across the graph analytics lifecycle.

Learn how Oracle is helping Toyota Mapmaster to accelerate digital transformation in map production

The Forrester Wave™: Graph Data Platforms, Q4 2020

Oracle is named a leader.

Download now

Introducing Oracle's graph database

See how Oracle's graph database makes it easy to explore relationships and discover connections in data by providing support for different graph structures, powerful analytics, and intuitive visualization.

Watch video (2:12)

17 use cases for graph databases and graph analytics

Discover graph use cases across industries and categories, including financial services, manufacturing, and machine learning research.

Download ebook

Why a graph database from Oracle?

Complete graph database

Oracle provides support for both property and RDF knowledge graphs, and simplifies the process of modeling relational data as graph structures. Interactive graph queries can run directly on graph data or in a high-performance in-memory graph server. Extensive integration with Oracle Database, Oracle Autonomous Database, and third-party and open-source features make it simpler to apply and use graph analytics.

Comprehensive graph analytics

Explore relationships with more than 60 prebuilt algorithms. Use SQL, native graph languages, Java and Python APIs, and Oracle Autonomous Database features to create, query, and analyze graphs. Then, display connections easily in data to discover insights like customer trends and fraud detection, and then use interactive tools to publish and share analysis results.

Enterprise-level scalability and security

Gain fine-grained security, high availability, easy manageability, and integration with all other data in business applications. Oracle provides sophisticated, multilevel access control for property graphs vertices and edges, and RDF triples. Oracle also aligns with applicable ISO and Worldwide Web Consortium standards for representing and defining graphs and graph query languages.

Graph database and graph technologies

Graph Studio in Oracle Autonomous Database

Graph in Oracle Database

Graph Studio in Oracle Autonomous Database

With Graph Studio, almost anyone can get started with graphs to explore relationships in data. Graph Studio removes barriers to


ORACLE