ORACLE

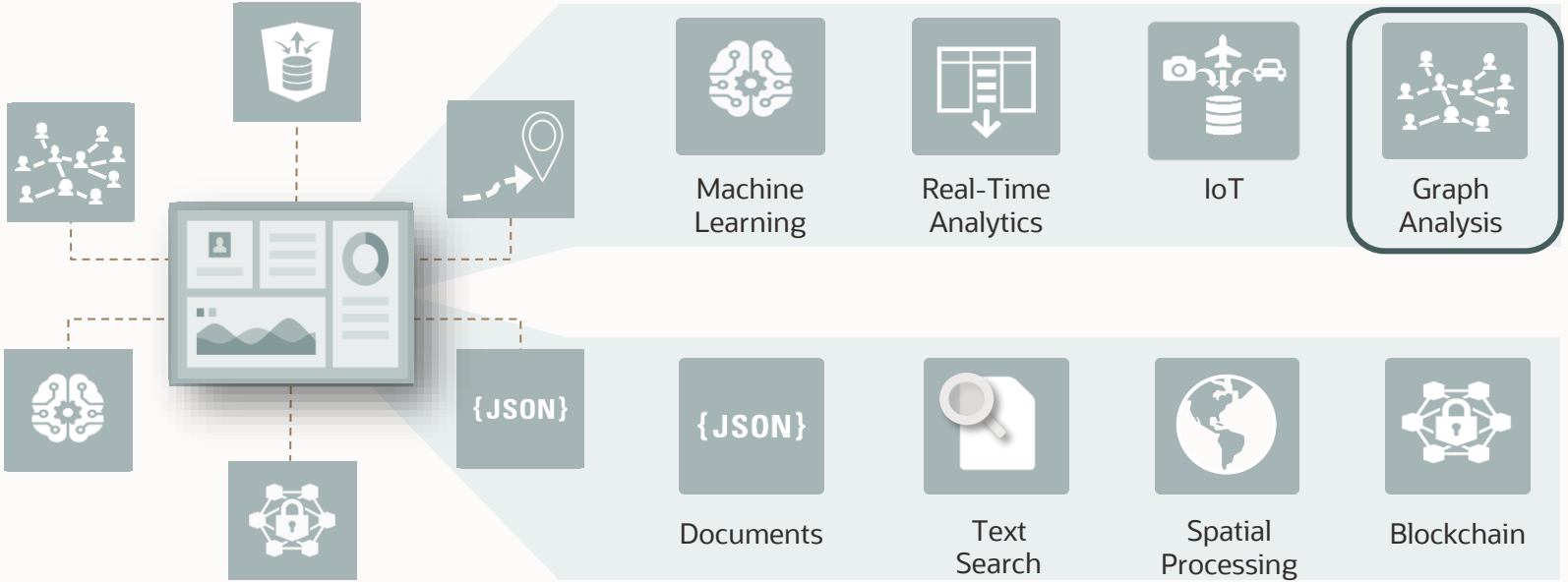
Getting started with Property Graphs

A Jumpstart for Database Developers

Hans Viehmann
Director Product Management – Spatial and Graph
nlOUG DB World 2026
November 13, 2026

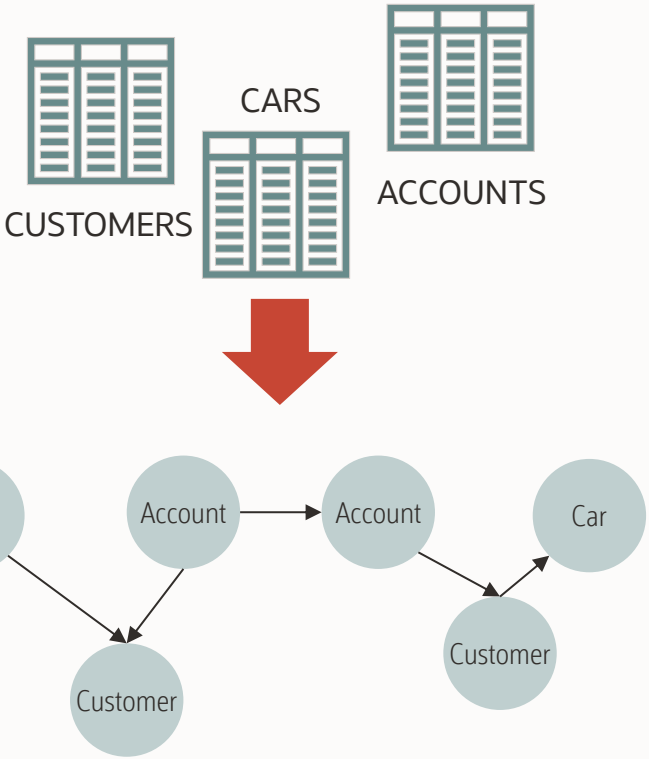


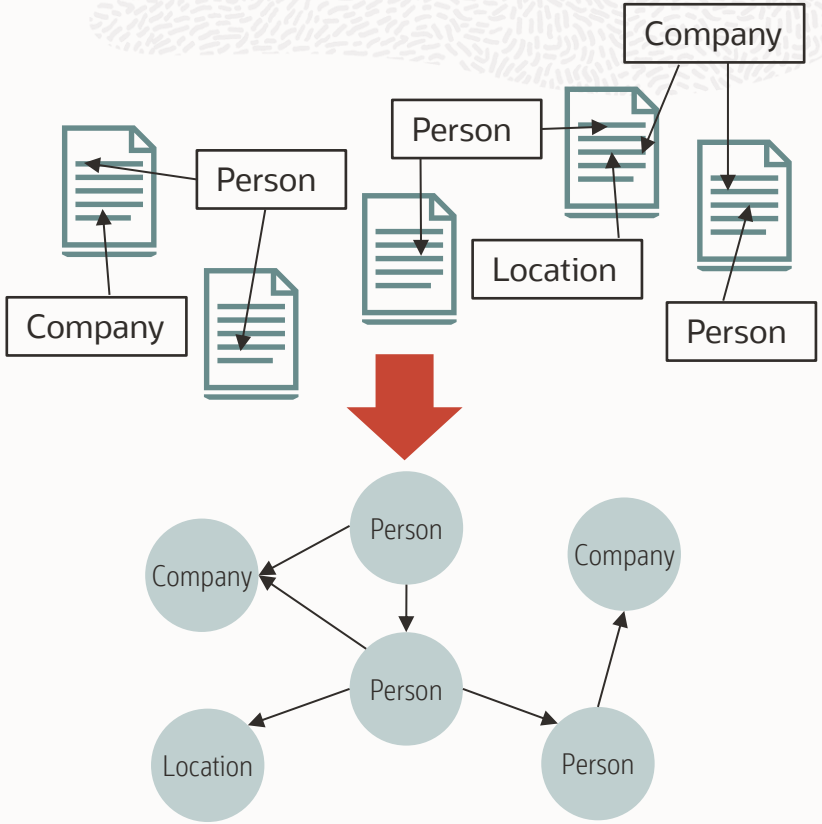
Modern Apps Need To Generate Value From Data in New Ways



Copyright © 2026, Oracle and/or its affiliates.

Graph Technologies

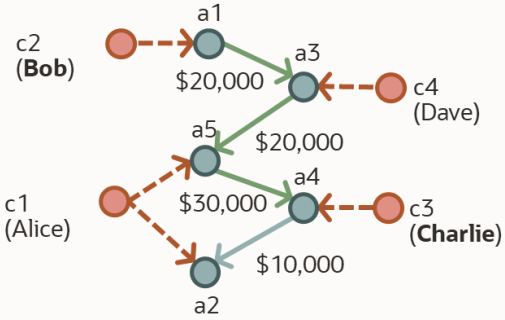




Copyright © 2026, Oracle and/or its affiliates.

How can you analyze graphs?

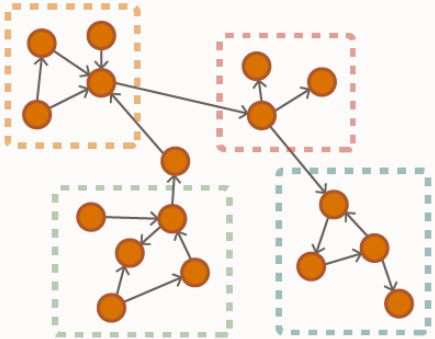
*Is there **any money flow** between Bob & Charlie?*



Graph-Based Pattern-Matching Queries:

- Fast traversals
- Path finding
- Identify patterns
- Extract subgraphs

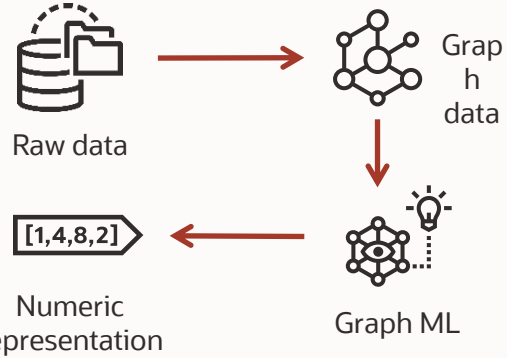
*How to identify the **clusters** of transactions?*



Graph Algorithms:

- Community Detection
- Ranking and Centrality
- Paths and Connectivity
- Link Prediction, Similarity

*How to detect **similar structures**?*



Graph Machine Learning:

- Unsupervised vector representations of nodes, edges, graphs
- Graph-based supervised classification and regression
- Anomaly detection

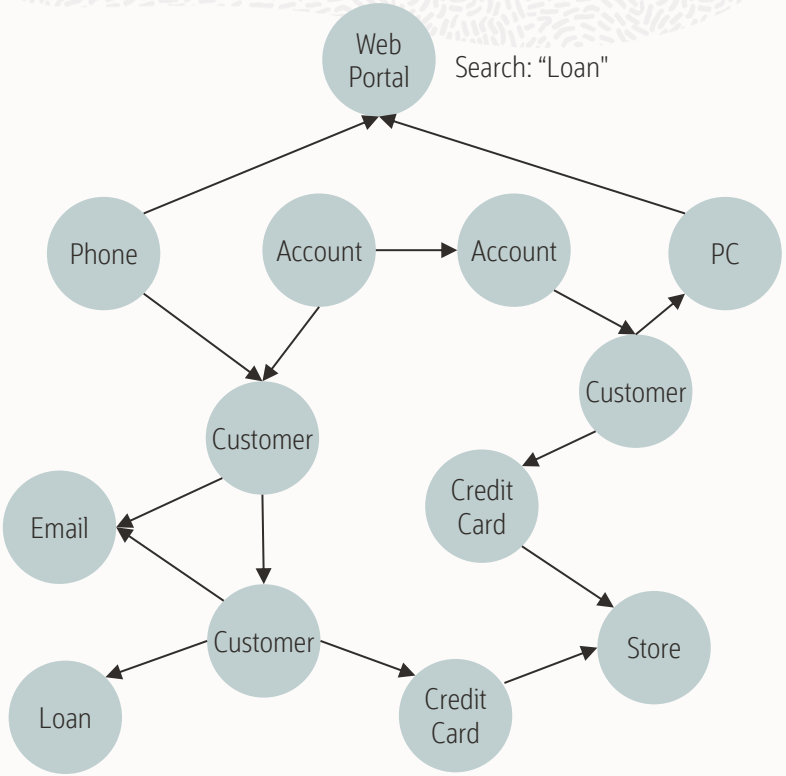
Copyright © 2026, Oracle and/or its affiliates.

Use case: Customer 360 Analysis in e-Commerce

Organize customer data in a data lake or data warehouse

Analyze relationships through graph algorithms, e.g.

- Associate entities and disambiguate data ("Entity resolution")
- Identify important entities and measure influence
- Determine communities for targeted marketing or risk management
- Predict links, e.g. for product recommendation



Copyright © 2026, Oracle and/or its affiliates.

Use case: BOM (Bill-of-material) Analysis in Manufacturing

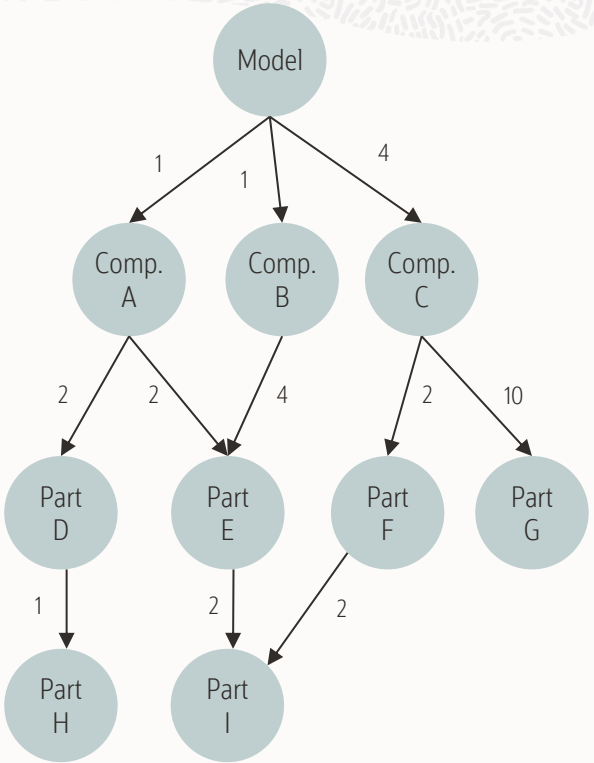
Manage complex hierarchies of parts and components

Analyze nested relationships using very compact queries, e.g.

- Aggregate required parts for a product variant
- Analyze dependency on a given part

Similar use cases:

- Data lineage
- GDPR compliance



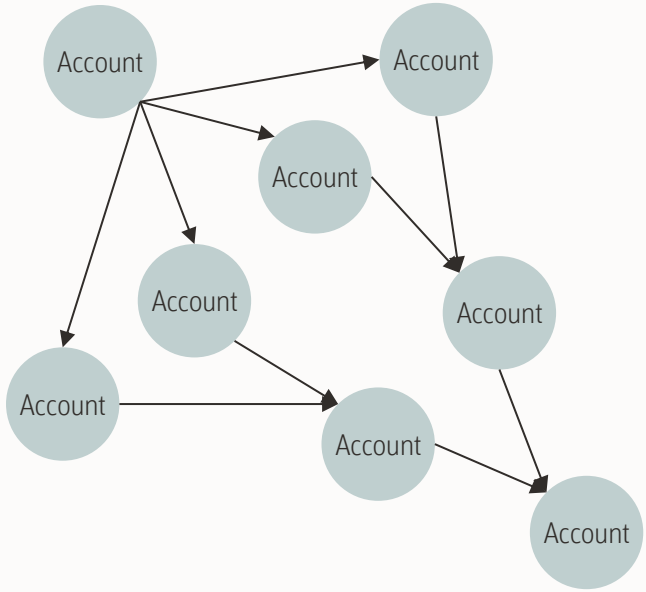
Copyright © 2026, Oracle and/or its affiliates.

Use case: Fraud Prevention in Financial Services

Manage large networks of interconnected entities

Detect suspicious patterns in financial transactions, e.g.

- Indirect connections to suspicious/fraudulent accounts
- Circular money flows, indicative of money laundering
- Layering (hiding money flows in large number of small, indirect transactions)



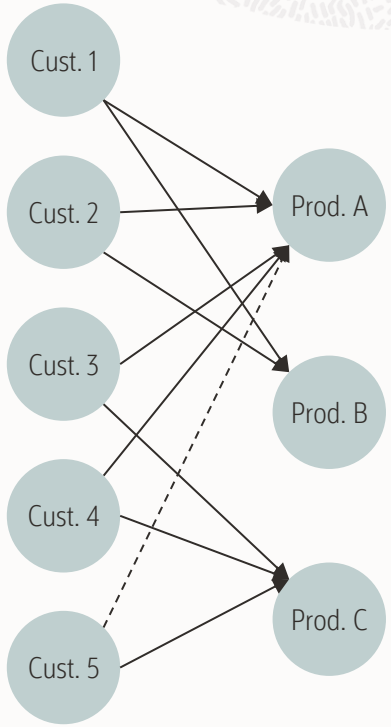
Copyright © 2026, Oracle and/or its affiliates.

Use case: Link Prediction for Product Recommendation

Predict relationships based on existing relationships

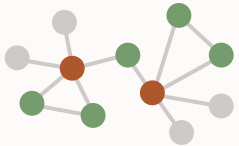
Graph analysis can provide real-time results

- Working on up-to-date data
- Taking into account strength and quality of indirect relationships
- Improving accuracy

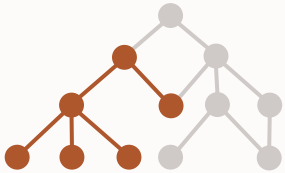


Copyright © 2026, Oracle and/or its affiliates.

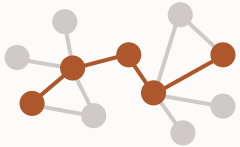
Analyzing relationships, direct or indirect



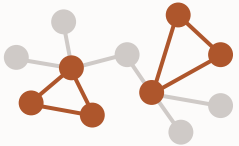
Networks



Hierarchies

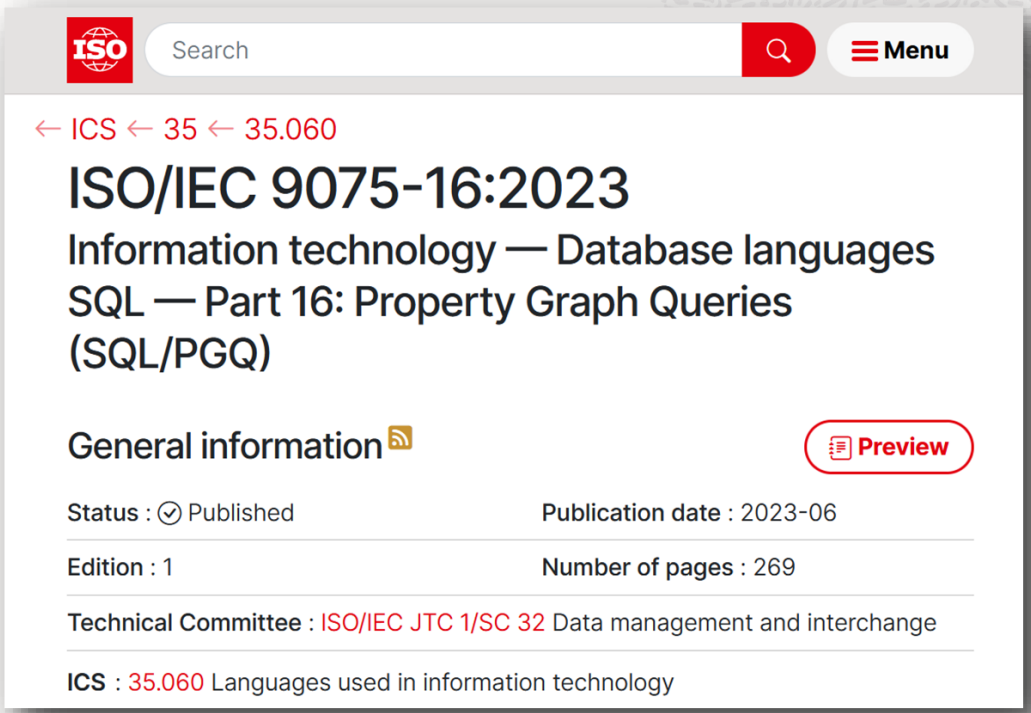


Paths



Patterns

Copyright © 2026, Oracle and/or its affiliates.



Copyright © 2026, Oracle and/or its affiliates.

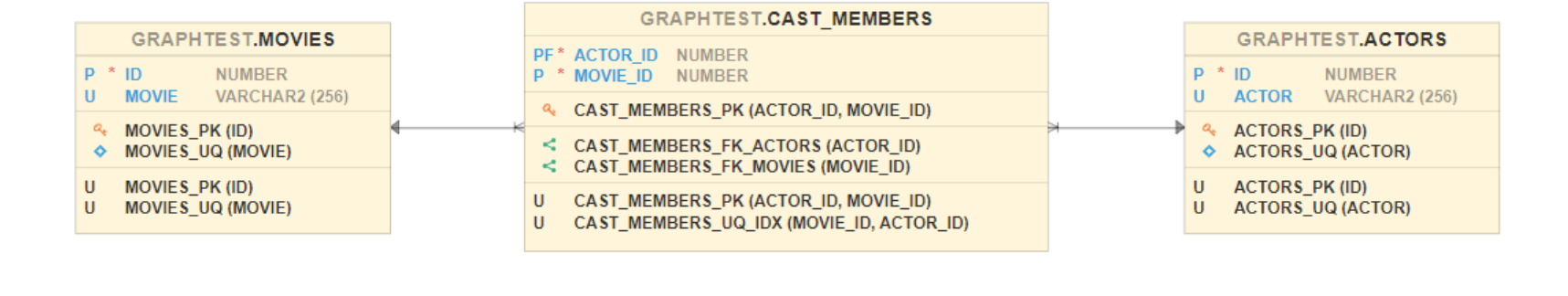
Example

Six degrees of Kevin Bacon - https://github.com/kibeha/ways-to-bacon

Copyright © 2026, Oracle and/or its affiliates.

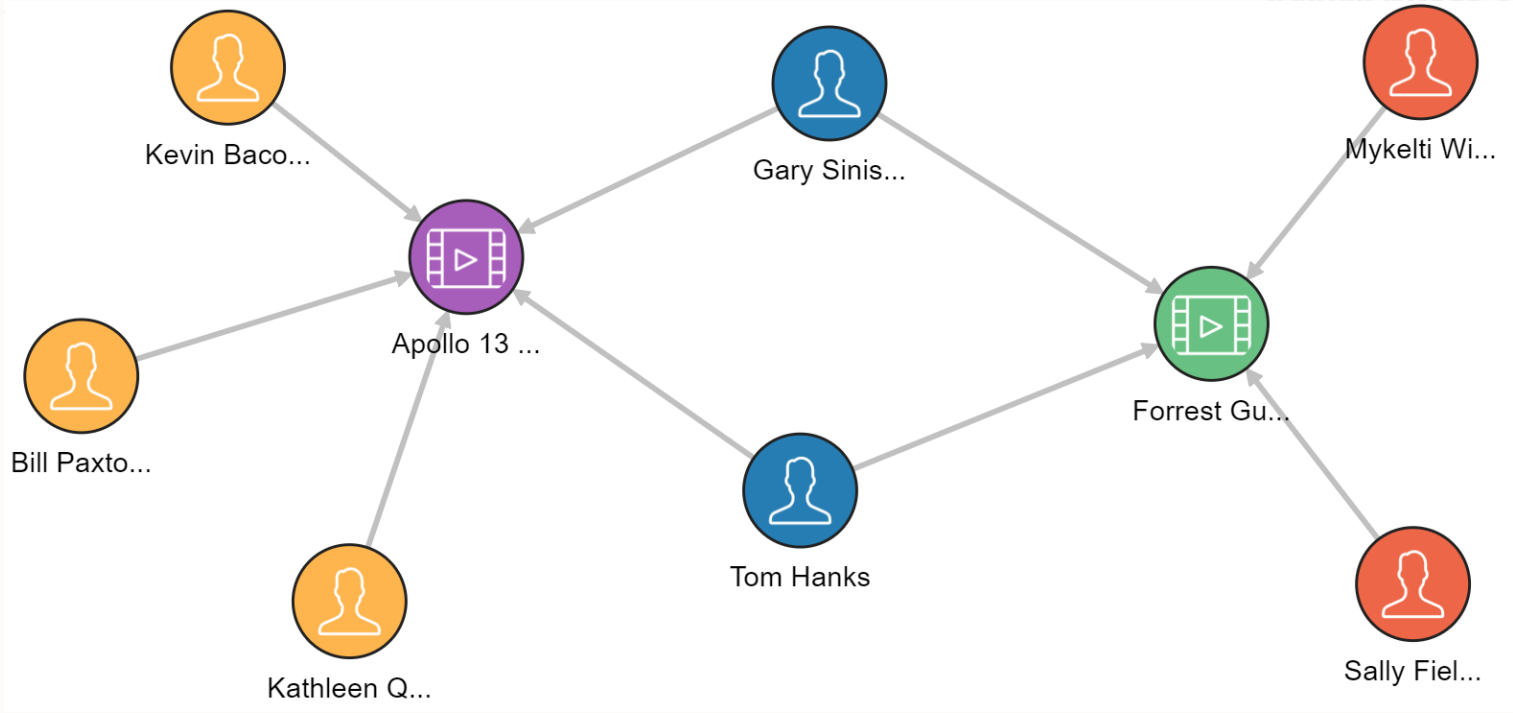


Starting point – a relational data model



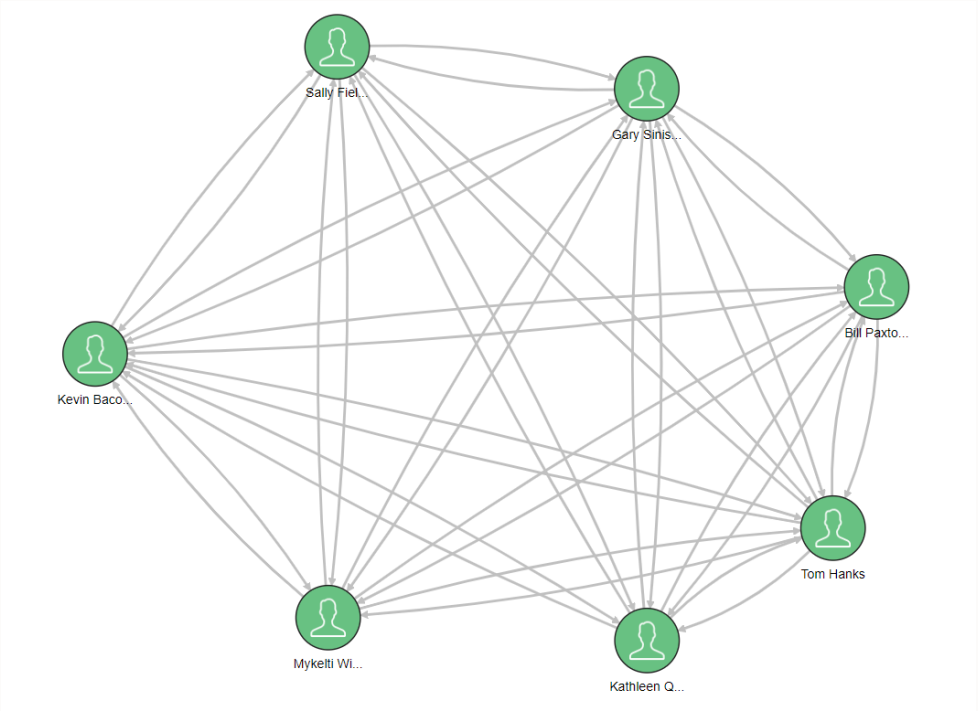
Copyright © 2026, Oracle and/or its affiliates.

Modeling the tables as a graph



Copyright © 2026, Oracle and/or its affiliates.

Modeling the tables as a graph – alternative model



Copyright © 2026, Oracle and/or its affiliates.

Mapping relational tables to a graph

More than one mapping possible

- Model depends on planned queries and analyses

Different from relational model

- N:m relationships are no issue

Definition of graph in PGQL DDL

- VERTEX TABLES clause and EDGE TABLES clause list the source tables
- Defaults are taken from Oracle Data Dictionary

CREATE PROPERTY GRAPH creates metadata only

- No physical data copy

```
CREATE PROPERTY GRAPH MOVIES
  VERTEX TABLES (
    actors
      KEY ( id )
      LABEL actor PROPERTIES ( name ),
    movies
      KEY ( id )
      LABEL movie PROPERTIES ( title )
  )
  EDGE TABLES (
    cast_members KEY ( actor_id, movie_id )
      SOURCE KEY ( actor_id ) REFERENCES actors
      DESTINATION KEY ( movie_id ) REFERENCES movies
      LABEL cast_member NO PROPERTIES
  )
```


Copyright © 2026, Oracle and/or its affiliates.

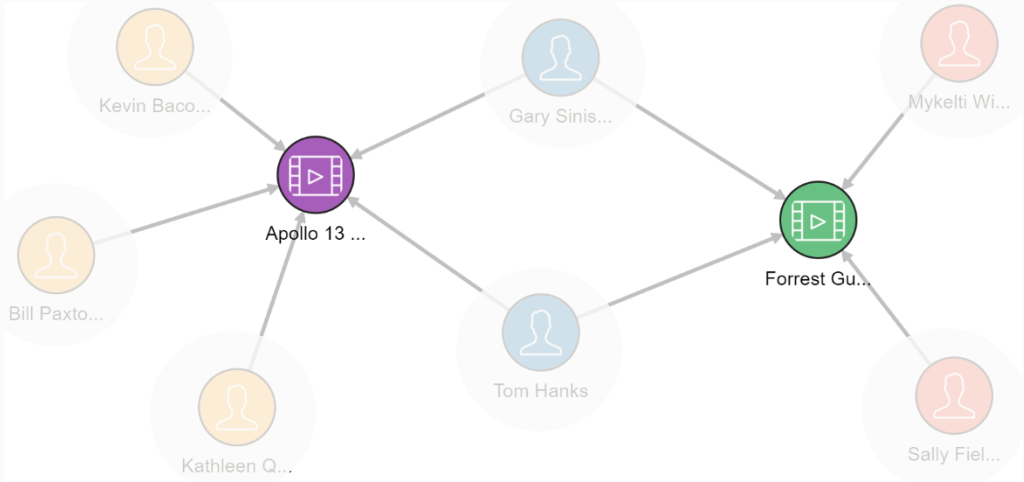
Demo

Graph modeler



Pattern matching on graphs

Using the MATCH clause

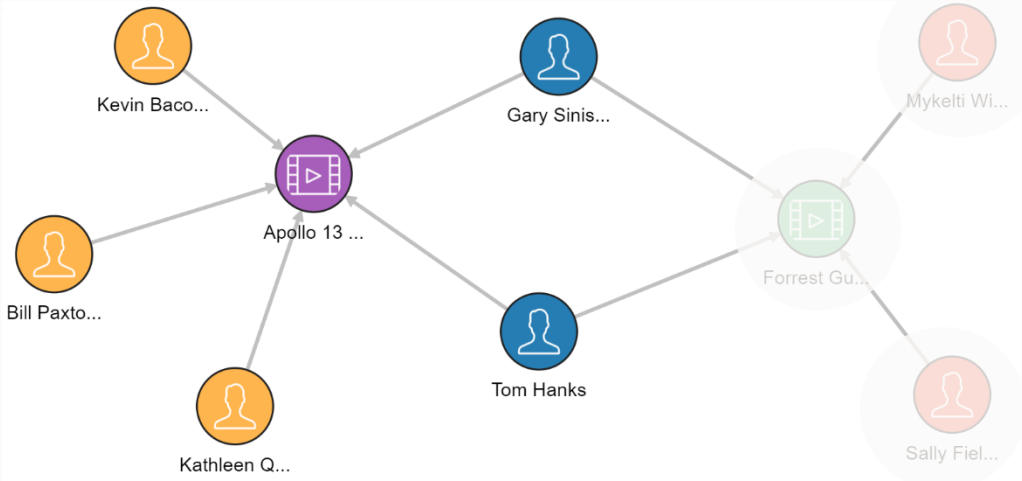



```
MATCH (m is movie)
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

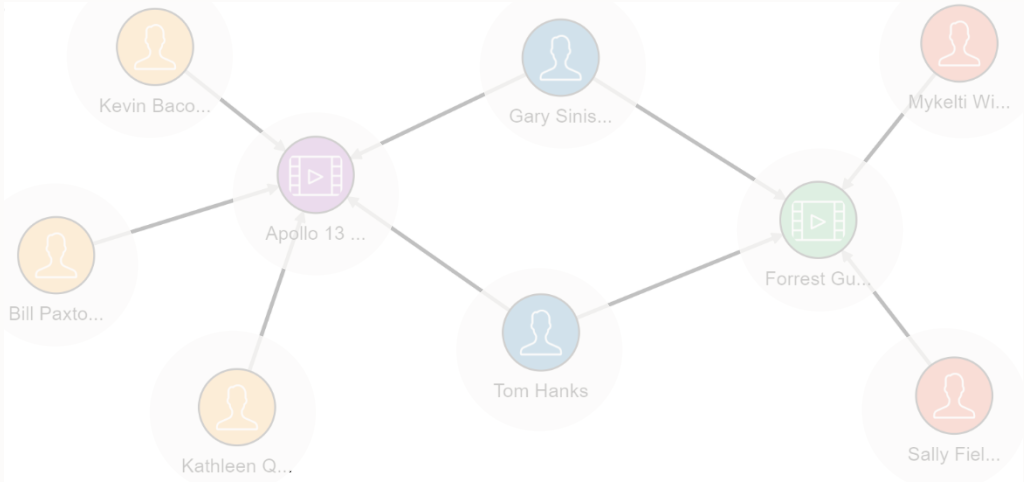



```
MATCH (a is actor) -[e]-> (m is movie)
WHERE m.title = 'Apollo 13'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

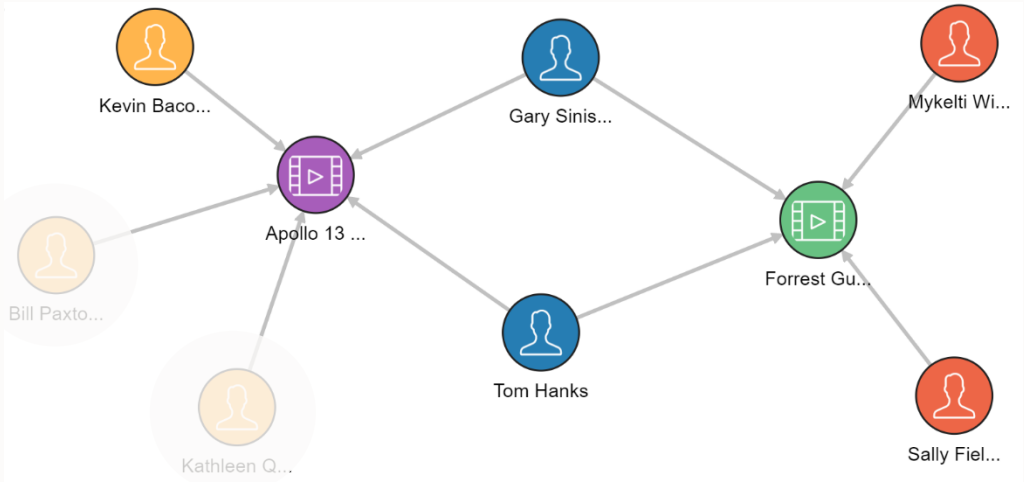



```
MATCH (a is actor) <-[e]- (m is movie)
WHERE m.title = 'Apollo 13'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

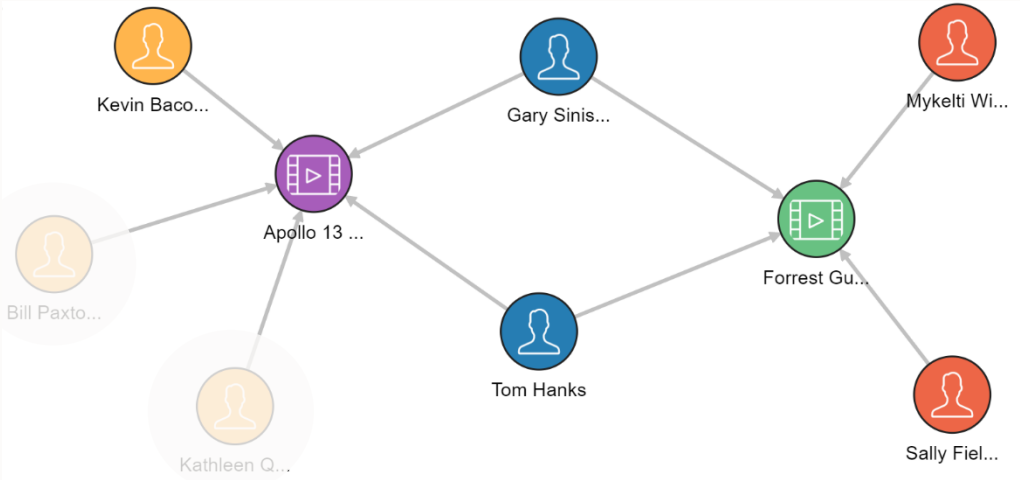



```
MATCH (a is actor) –[e1]-> (m1 is movie)
         <-[e2]- (a2 is actor) –[e3]->
      (m2 is movie) <-[e4]- (a3 is actor)
WHERE a.name=’Kevin Bacon’ AND m2.title=’Forrest Gump’
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

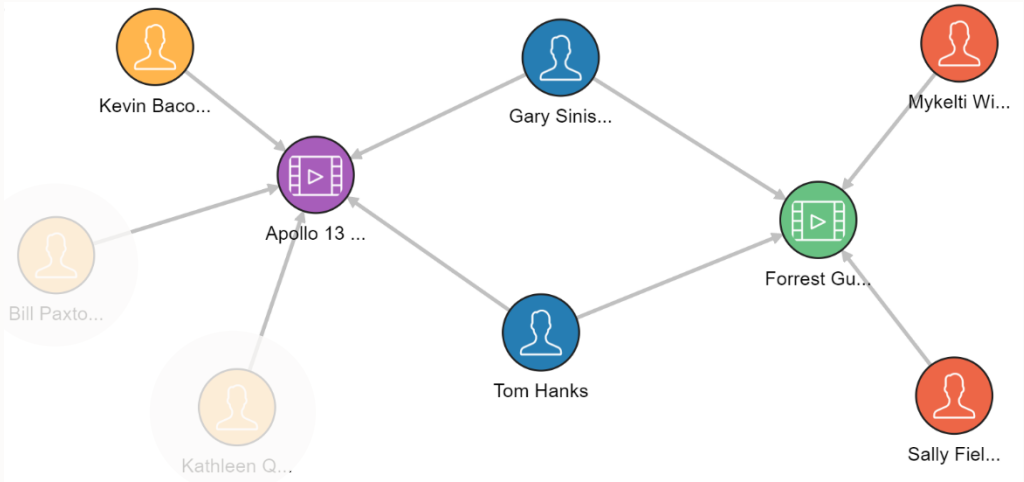



```
MATCH (a is actor) -[e1]-> (m1)
            <-[e2]- (a2) -[e3]->
             (m2) <-[e4]- (a3)
WHERE a.name='Kevin Bacon' AND m2.title='Forrest Gump'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

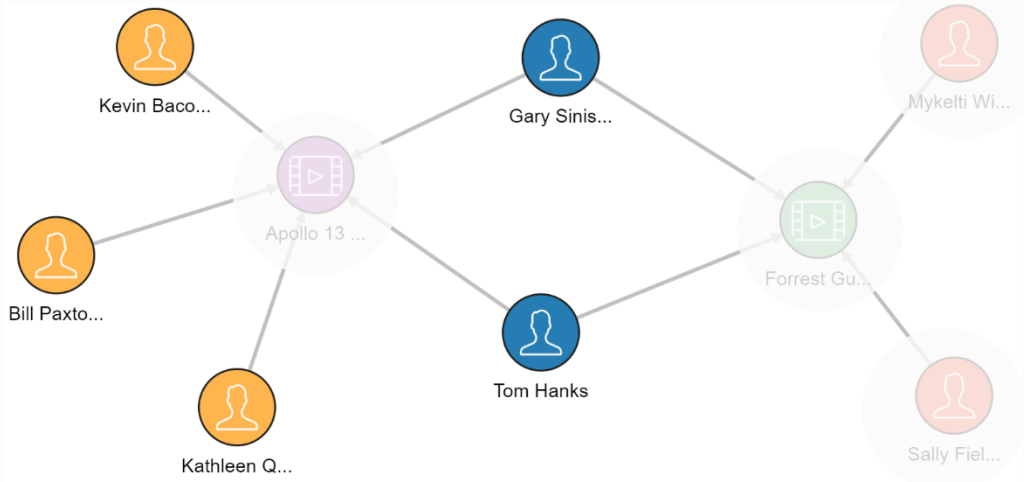



```
MATCH (a is actor) -[e1]- (m1)
            -[e2]- (a2) -[e3]-
            (m2) -[e4]- (a3)
WHERE a.name='Kevin Bacon' AND m2.title='Forrest Gump'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

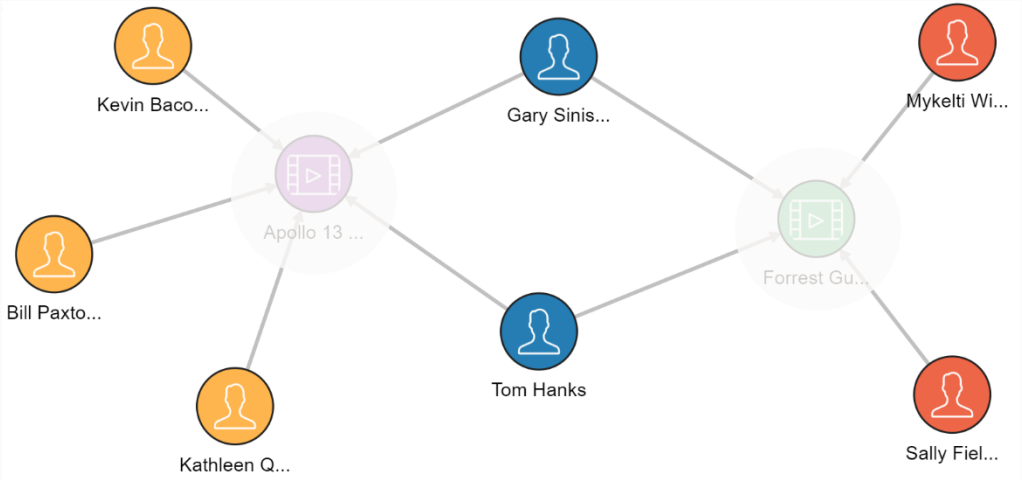



```
MATCH (a is actor)
      -[e]-{2} (a2)
WHERE a.name='Kevin Bacon'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

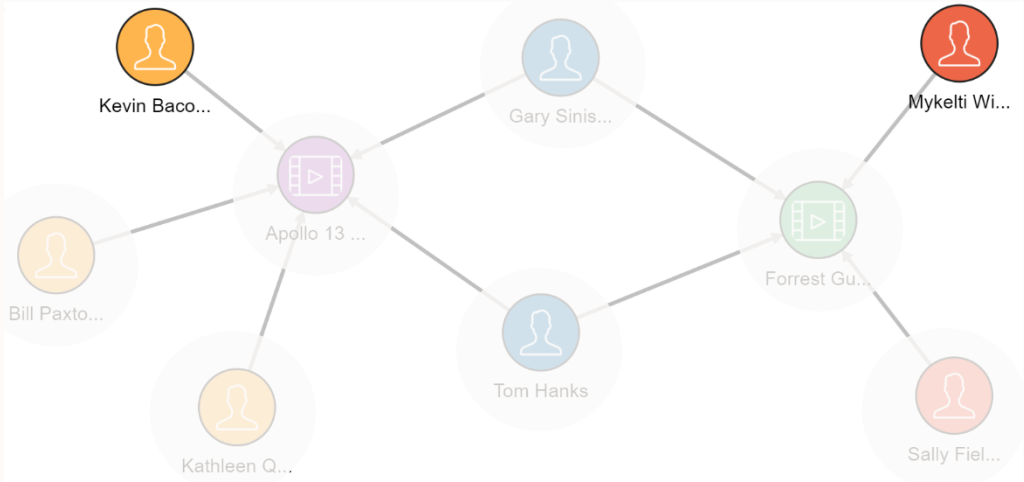



```
MATCH (a is actor)
      -[e]-{4} (a2)
WHERE a.name='Kevin Bacon'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

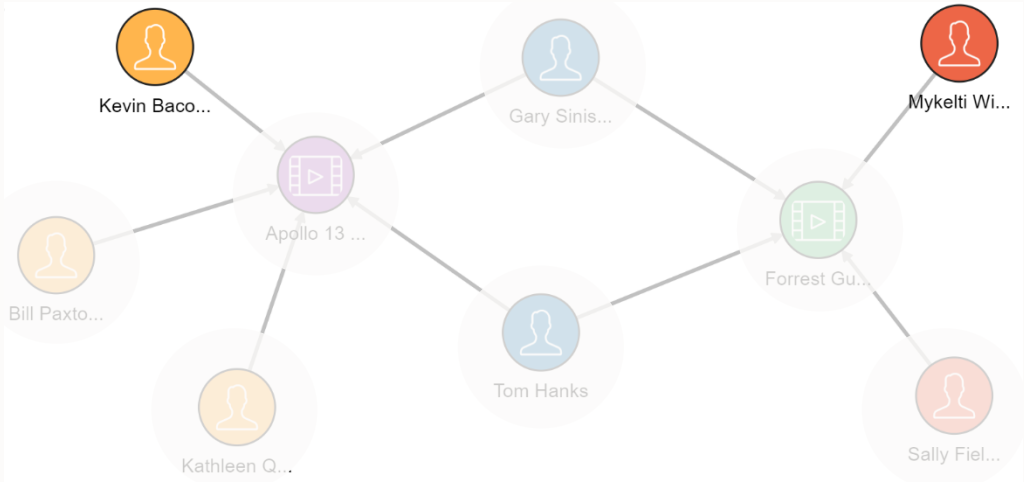



```
MATCH (a is actor)
      -[e]-{4} (a2)
WHERE a.name='Kevin Bacon' AND a2.name='Mykelti Williamson'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

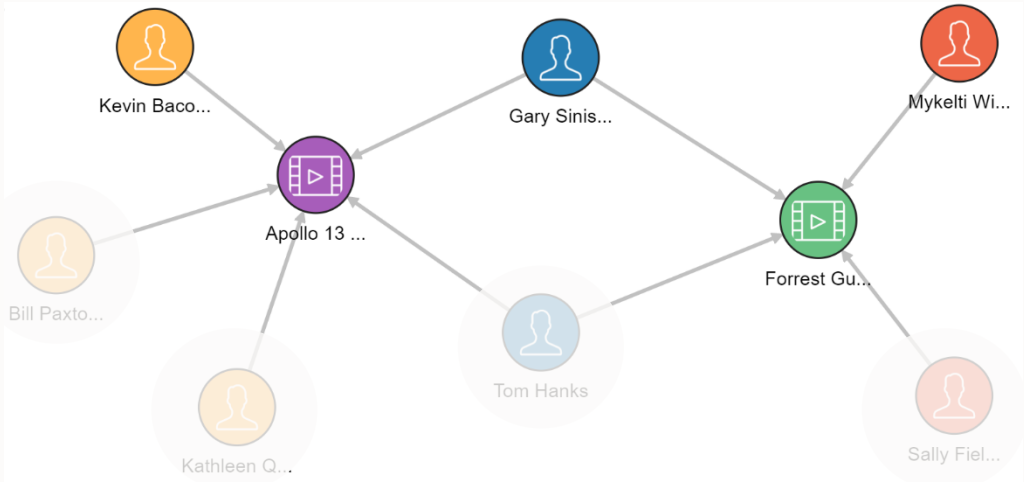



```
MATCH SHORTEST (a is actor)
      -[e]-* (a2)
WHERE a.name='Kevin Bacon' AND a2.name='Mykelti Williamson'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause

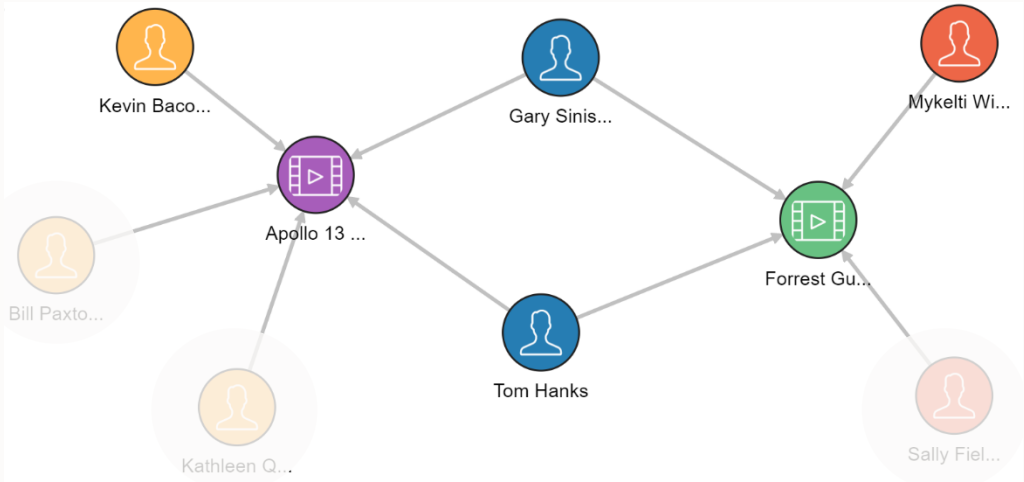



```
MATCH SHORTEST (a is actor)
      (-[e]-(s))* (a2)
WHERE a.name='Kevin Bacon' AND a2.name='Mykelti Williamson'
```


Copyright © 2026, Oracle and/or its affiliates.

Pattern matching on graphs

Using the MATCH clause




```
MATCH ALL SHORTEST (a is actor)
      (-[e]-(s))* (a2)
WHERE a.name='Kevin Bacon' AND a2.name='Mykelti Williamson'
```


Copyright © 2026, Oracle and/or its affiliates.

More pattern matching language constructs

Shortest path/cheapest path

- Any, all, or top <k> shortest paths
- Cheapest path uses cost function
- Option to filter out paths with repeating vertices or edges (ACYCLIC, TRAIL, …)

```
MATCH ALL ACYCLIC PATHS (src) -[e]->+ (dst)
```

```
MATCH ANY TRAIL (src) (-[e]->(hop)){5,9} (dst)
```

```
MATCH SHORTEST 3 PATHS  (src) -[e]->* (dst)
```

```
MATCH ANY CHEAPEST PATH (src)
                -[e COST e.amount]->* (dst)
```

Resolving paths

- ONE ROW PER STEP retrieves vertices and edges along path
- ELEMENT_NUMBER() returns the position of the vertex or edge along the path

```
MATCH ANY (src) -[e]->+ (dst)
ON mygraph
ONE ROW PER STEP (v1, e1, v2)
```


Copyright © 2026, Oracle and/or its affiliates.

Using the MATCH clause

In SQL, the MATCH clause requires a GRAPH_TABLE function:

```
SELECT * FROM GRAPH_TABLE( movies
MATCH (a is actor) -[e]-> (m is movie)
WHERE m.title = 'Apollo 13'
COLUMNS (a.name, m.title) )
```

Notes:

- All variables in the COLUMNS clause must have different names
- WHERE clauses can be inside GRAPH_TABLE() or outside
- Group variables like repeating edges require aggregation
- Graphs currently work on tables and materialized views, not on views (yet)

Copyright © 2026, Oracle and/or its affiliates.



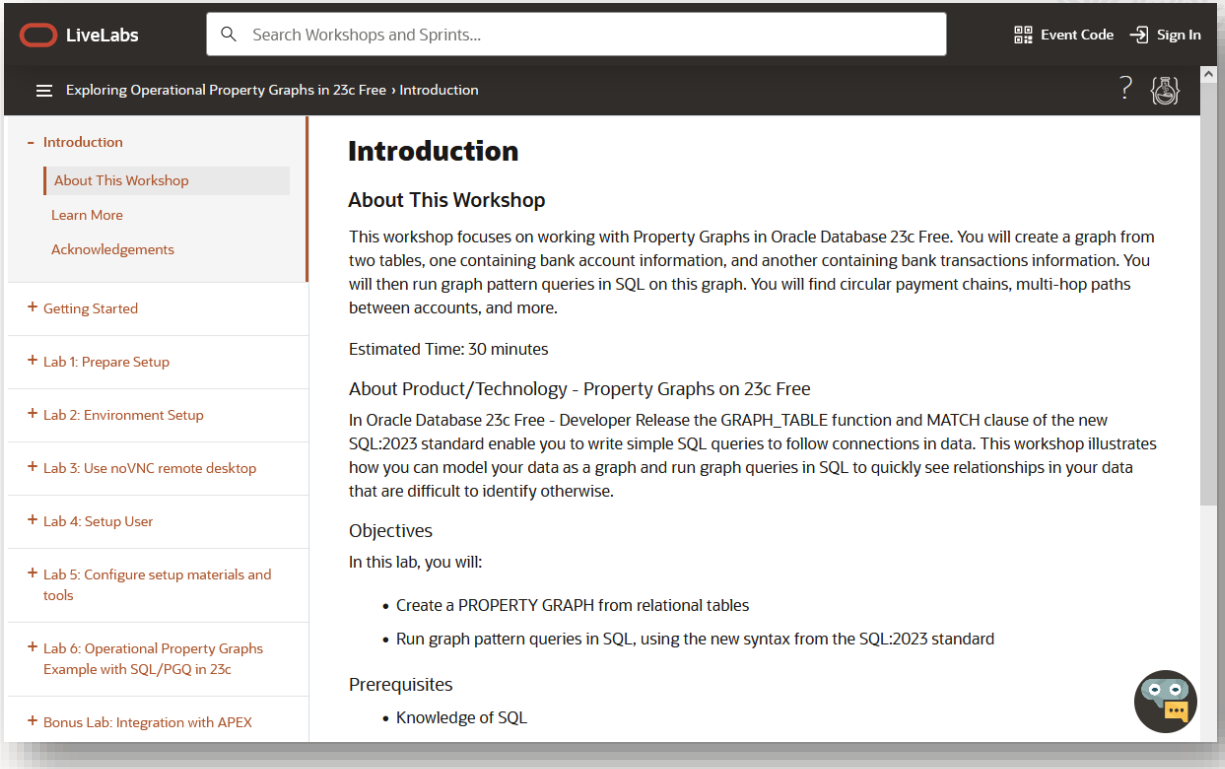
Demo

Visual graph query builder



Try it out yourself!

LiveLabs workshop “Exploring Operational Property Graphs” (here)

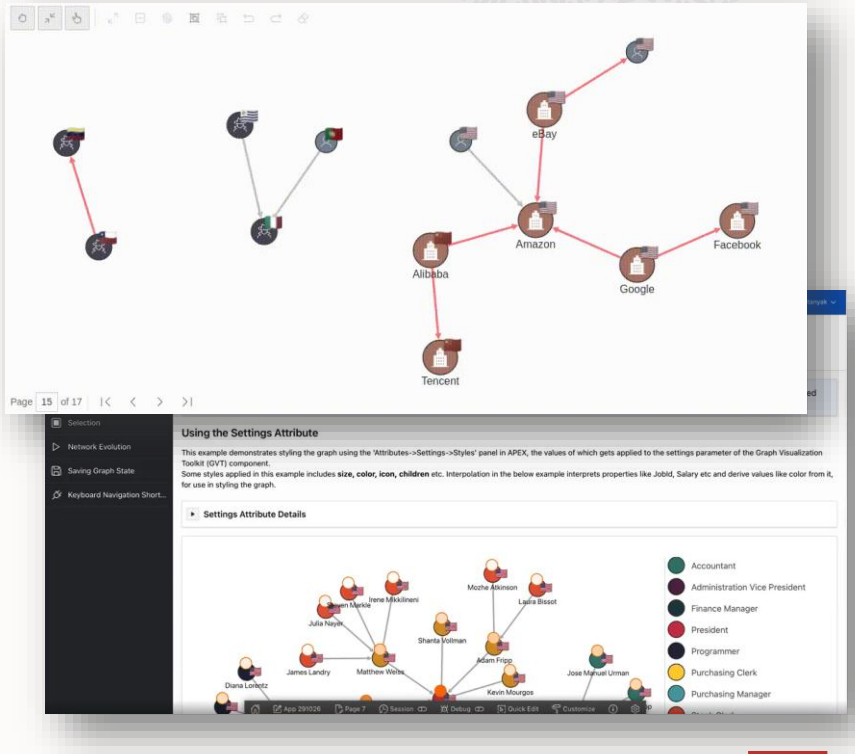


Copyright © 2026, Oracle and/or its affiliates.

APEX plug-in

Plug-in available in APEX plug-in repository

- Visualize graph query results in APEX
  - Using Graph Visualization Toolkit
- Explore the graph vertices and edges
- Interact with the graph visualization
  - changing graph layouts, grouping or ungrouping, removing selected vertices or edges, …
- Style vertices and edges (size, color, icon, label values, …)
- Visualize and study the evolution of the graph over time
- Sample application on oracle.github.io/apex/



Copyright © 2026, Oracle and/or its affiliates.

Summary

Graphs offer a different view on existing data

Using Oracle Graph simplifies application development

Oracle Graph is the graph database for the enterprise



Copyright © 2026, Oracle and/or its affiliates.

Further information on Oracle Graph

Oracle Graph technologies landing page on oracle.com

- www.oracle.com/database/graph

Videos on Youtube

- www.youtube.com/c/OracleSpatialandGraph

Oracle LiveLabs: Search for „graph“ on

- livelabs.oracle.com

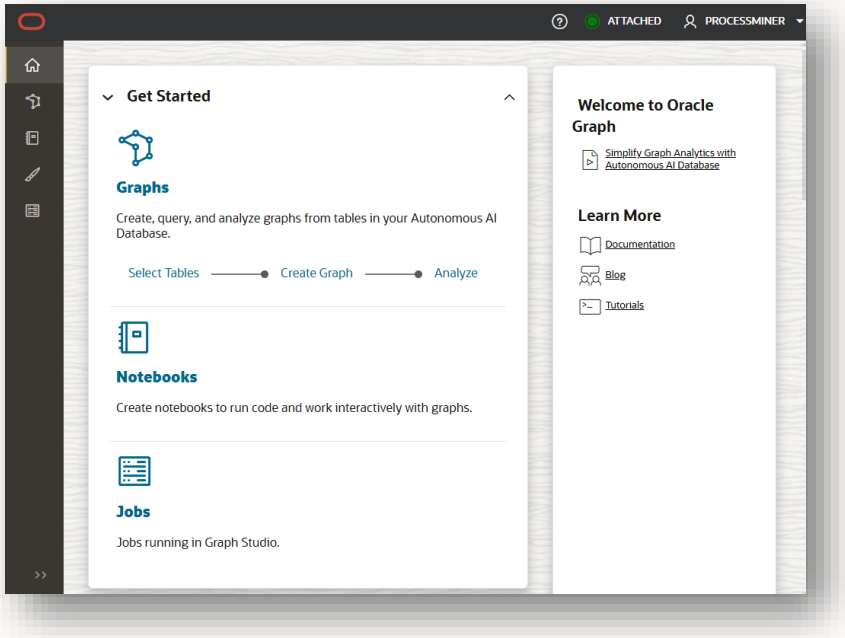
Blogs

- Database Insider (category: Graph)
- medium.com (tag: Oracle Graph)

Documentation

- Oracle Property Graph documentation
- Graph Studio in ADB documentation
- PG Viz Developers Guide

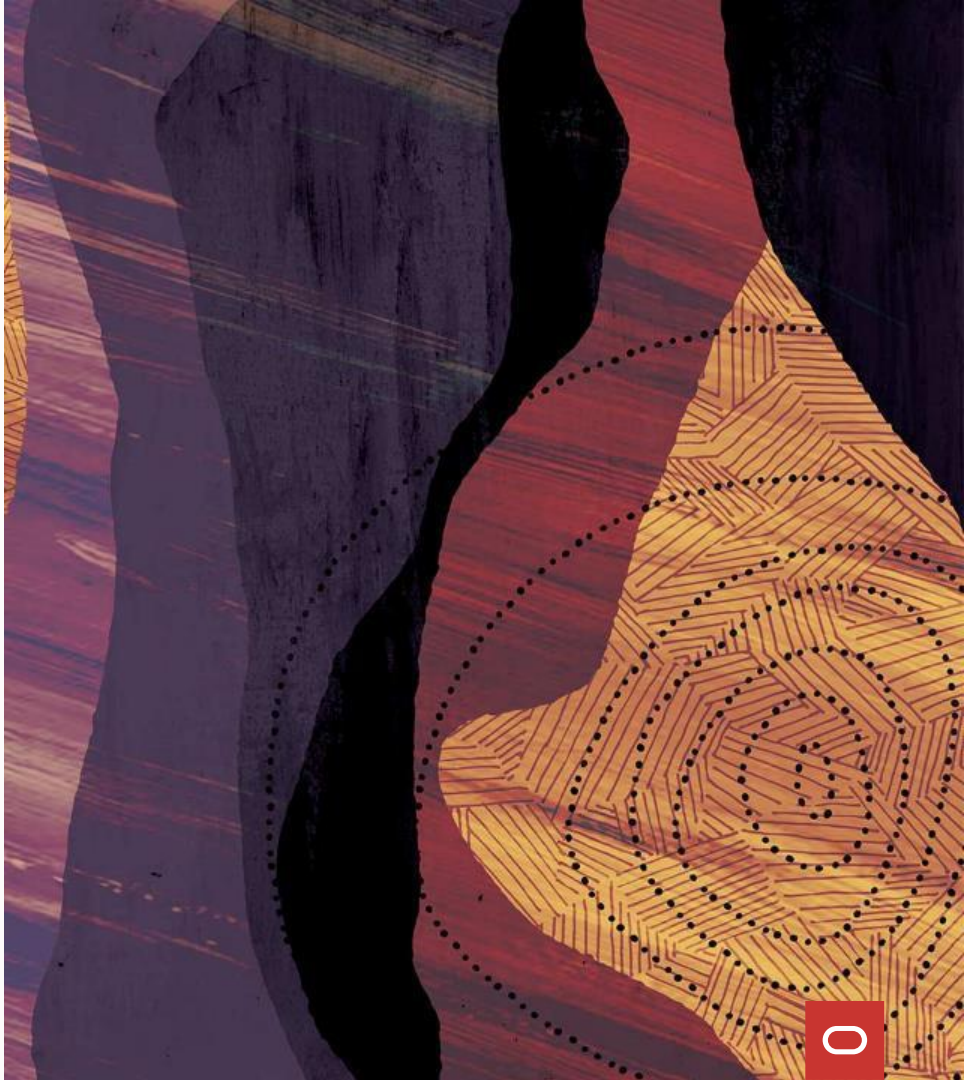
Skills repository on Github



Copyright © 2026, Oracle and/or its affiliates

Thank you

hans.viehmann@oracle.com

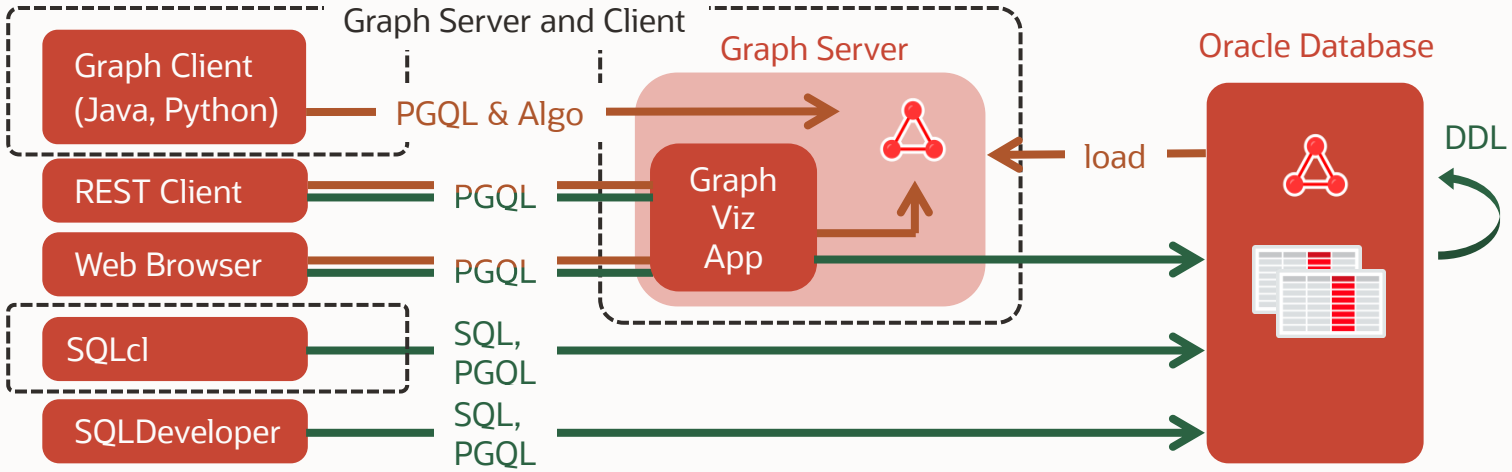


Copyright © 2026, Oracle and/or its affiliates.


ORACLE

Graph Server and Client

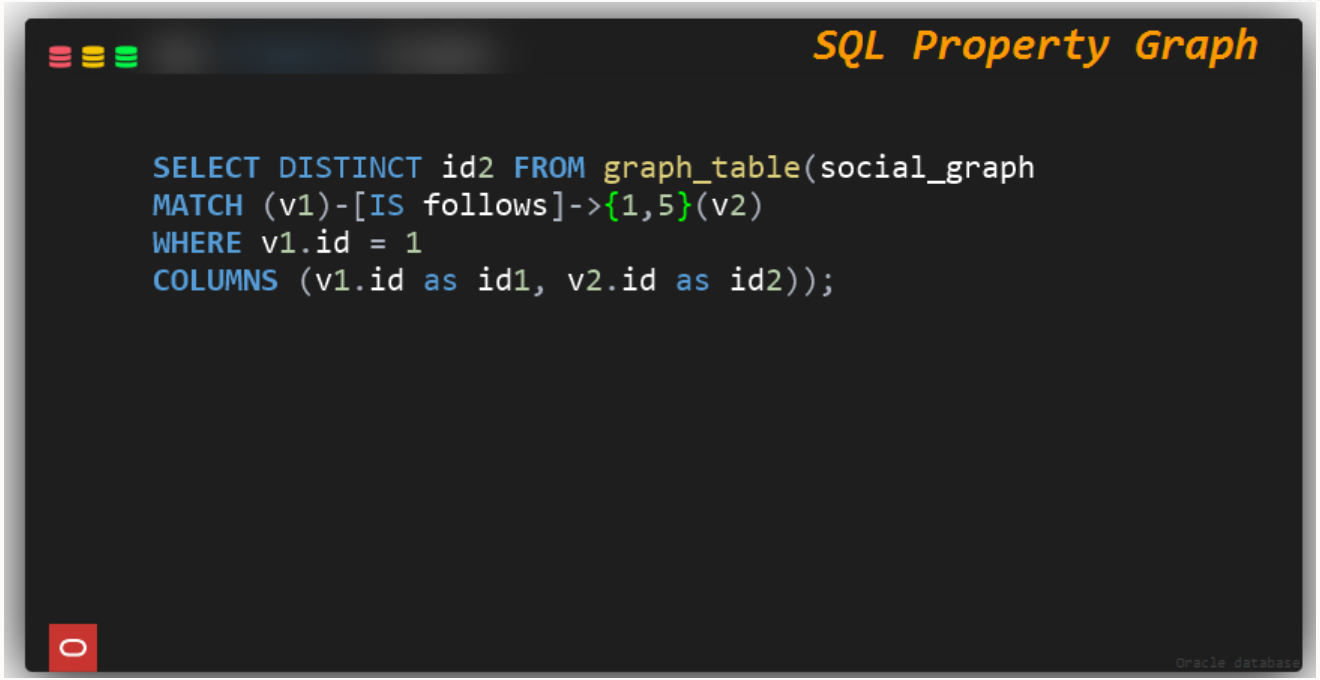
- Graph Server and Client is a component of the **Property Graph** feature of Oracle Database
- The latest Graph Server and Client works with Database 12.2 and later



Copyright © 2026, Oracle and/or its affiliates.

Benefits of pattern matching queries on graphs

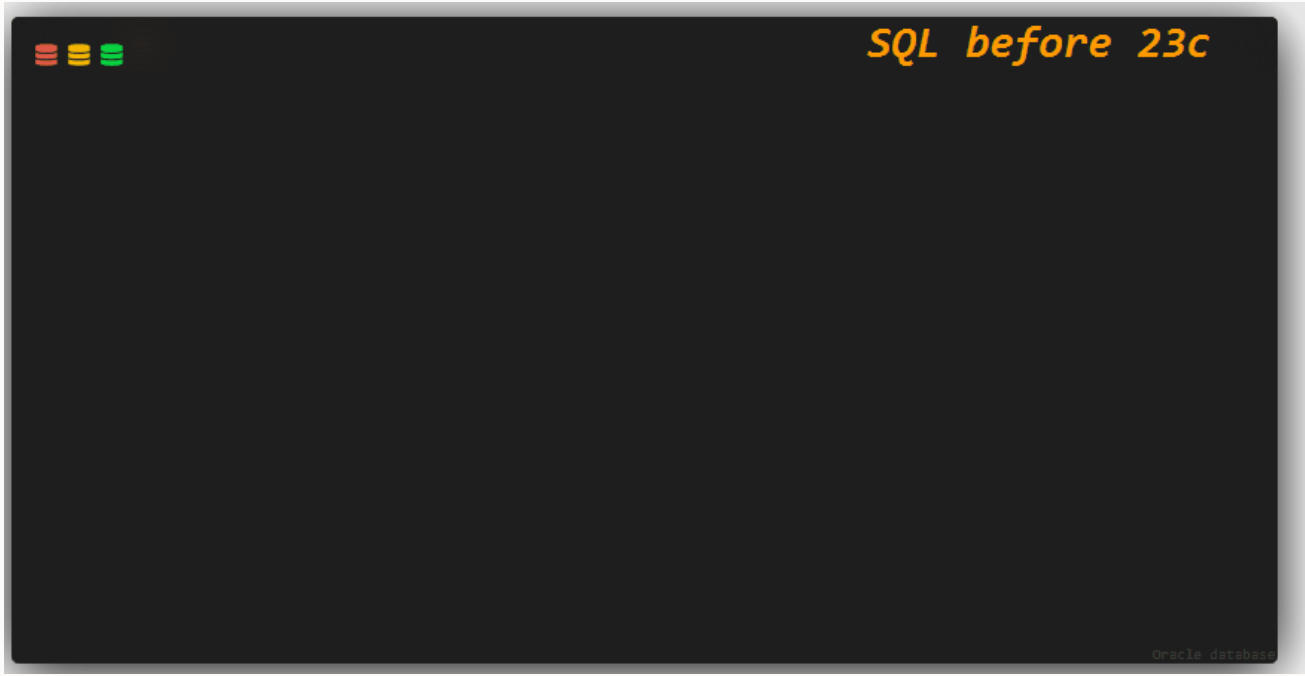
Analysis of indirect relationships = multi-hop queries



Copyright © 2026, Oracle and/or its affiliates.

Benefits of pattern matching queries on graphs

Benefits of pattern matching queries on graphs



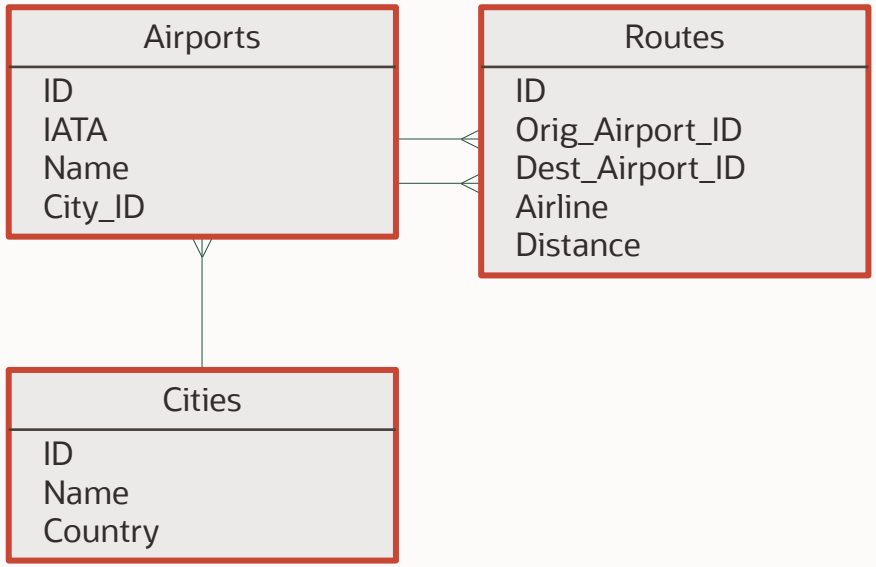
Copyright © 2026, Oracle and/or its affiliates.

Example

Travel analysis

Copyright © 2026, Oracle and/or its affiliates.

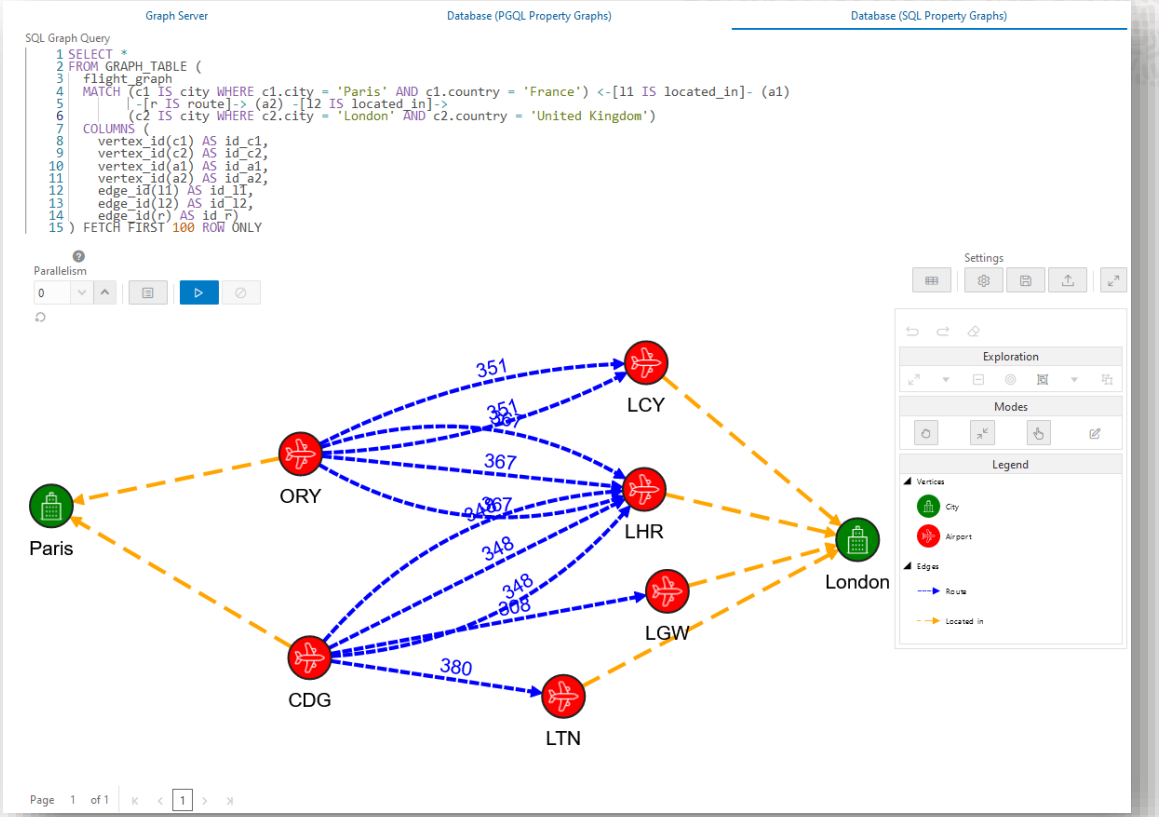
Mapping relational tables to the graph




```
CREATE PROPERTY GRAPH flights_graph
  VERTEX TABLES (
    airports
      LABEL airport
      PROPERTIES (name,iata),
    cities
      LABEL city
      PROPERTIES (city,country) )
  EDGE TABLES (
    routes
      SOURCE KEY (orig_airport_id)
          REFERENCES airports
      DESTINATION KEY (dest_airport_id)
          REFERENCES airports
      LABEL route
      PROPERTIES (airline,distance),
    airports AS airports2
      SOURCE KEY (city_id) REFERENCES cities
      DESTINATION KEY (id) REFERENCES airports
      LABEL located_in
      NO PROPERTIES )
```


Copyright © 2026, Oracle and/or its affiliates.

Graph model in GraphViz tool



Copyright © 2026, Oracle and/or its affiliates.

Demo

Travel analysis

Copyright © 2026, Oracle and/or its affiliates.

Recap: Using JSON documents with graphs

Support for JSON

- JSON document as property
- Element of JSON document as property

Example:

- TRAIN_CONNECTIONS table has DETAILS column of type JSON

Enables schema-flexible graphs

```
CREATE PROPERTY GRAPH flights_plus
  VERTEX TABLES (
    airports
      <...> ,
    cities
      <...> )
  EDGE TABLES (
    routes
      <...> ,
    airports AS airports2
      <...> ,
    train_connections AS tc
      SOURCE KEY (orig_airport_id) REFERENCES airports
      DESTINATION KEY (dest_airport_id) REFERENCES airports
      LABEL train_connection
      PROPERTIES (tc.details.Operator.string() as operator,
                  distance) )
```

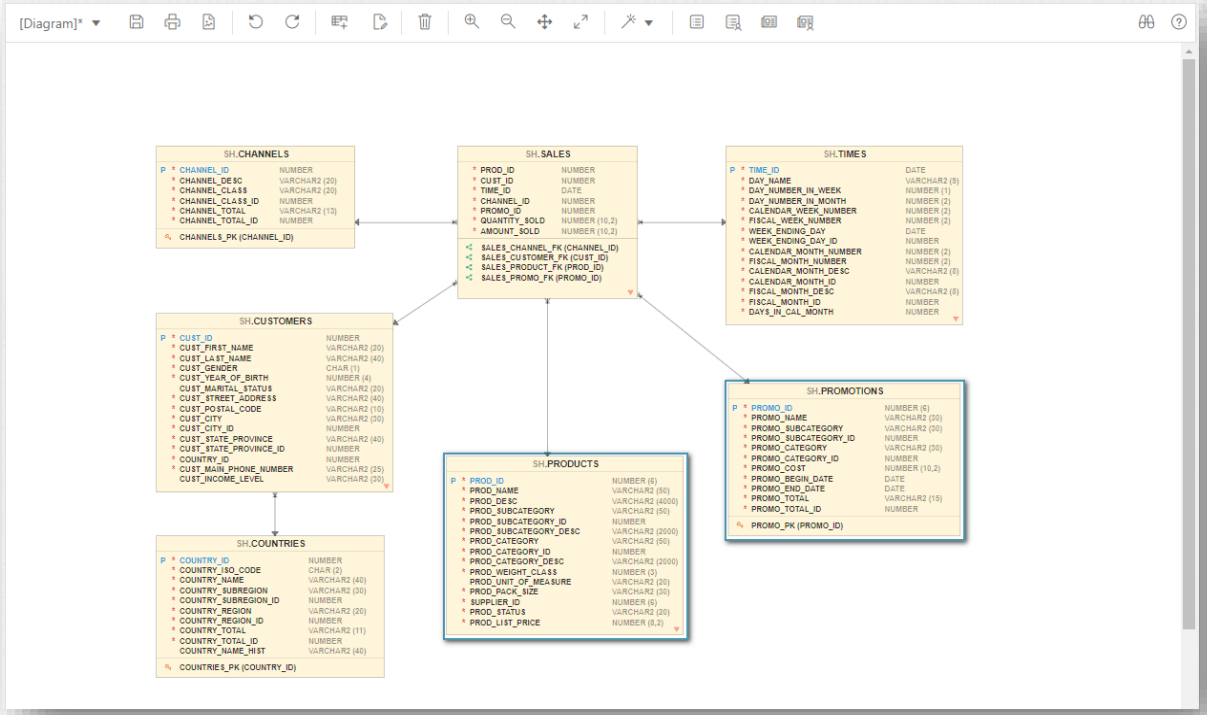

Copyright © 2026, Oracle and/or its affiliates.

Use case: Application modernization

Understanding data models and identifying bounded contexts

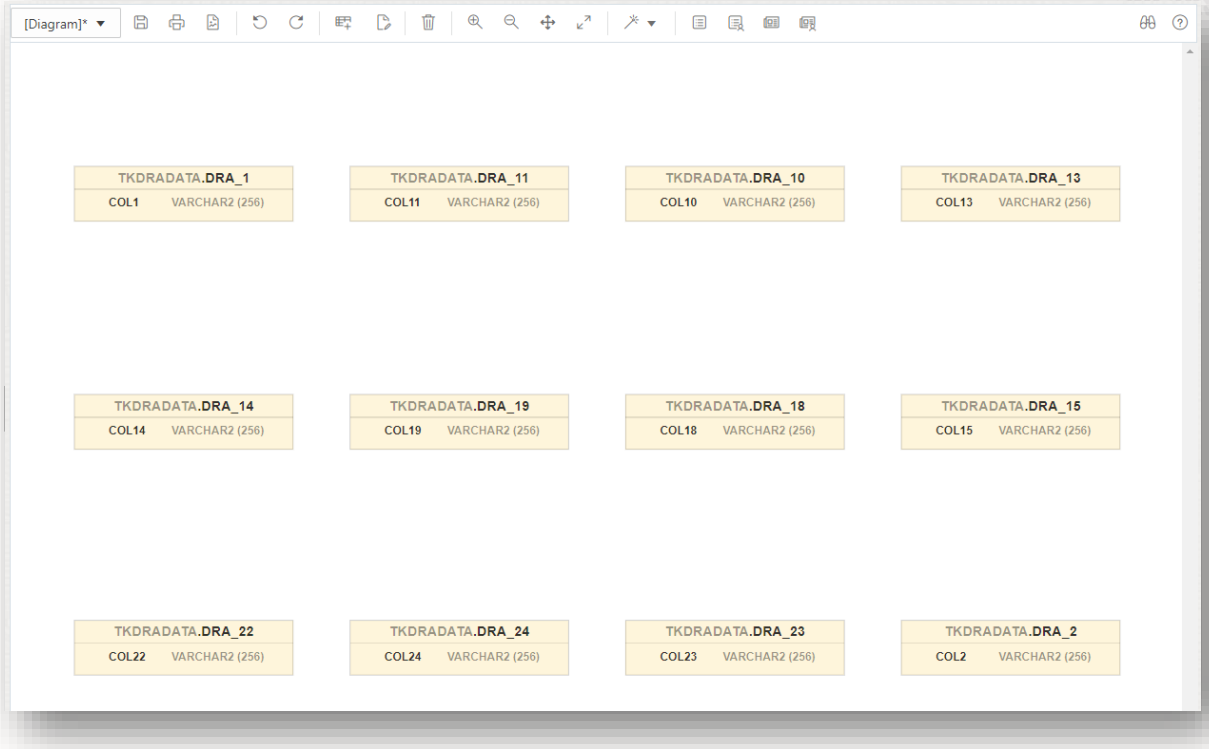
Copyright © 2026, Oracle and/or its affiliates.

In an ideal world ...



Copyright © 2026, Oracle and/or its affiliates.

In reality ...



Copyright © 2026, Oracle and/or its affiliates.

How to figure out what the application really does?

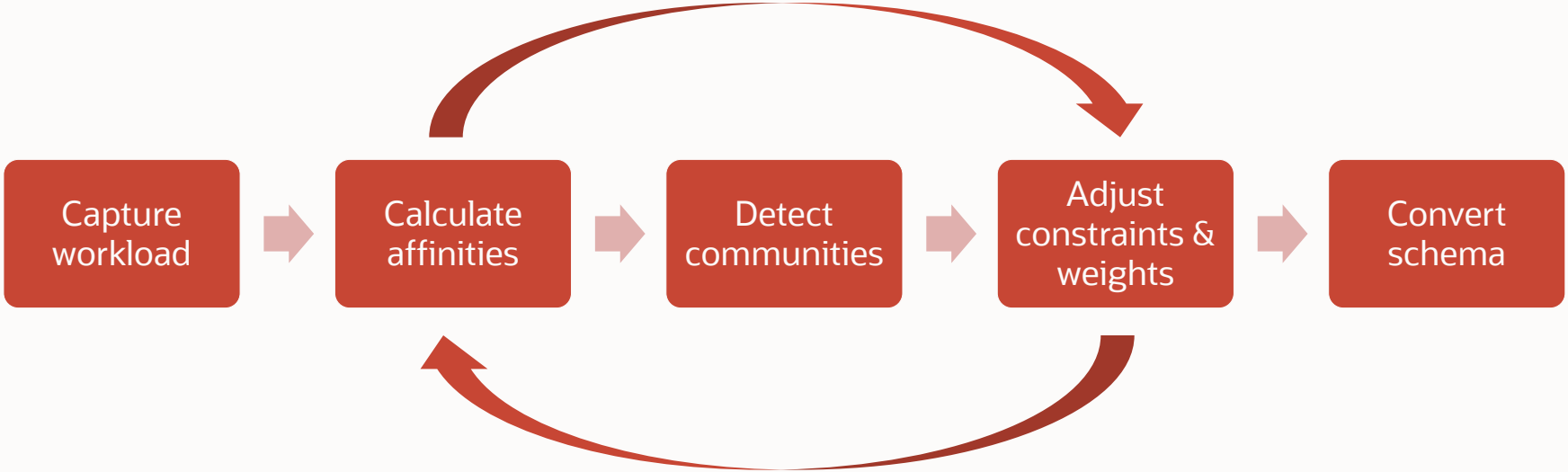
Create and analyze SQL Tuning set using DBMS_SQLTUNE or DBMS_SQLSET

1. Create SQL Tuning Set using CREATE_SQLSET
   - Group SQL statements and related metadata into one database object
   - Optionally export and analyze in non-production database
2. Capture all SQL statements from shared SQL area with SELECT_CURSOR_CACHE
   - Parsing schema, bind variables
   - Execution statistics
   - Execution plans
   - Row source statistics
3. Load results into Tuning set using LOAD_SQLSET
4. Analyze results in DBA_SQLSET_PLANS

Copyright © 2026, Oracle and/or its affiliates.

Using graphs for legacy software modernization

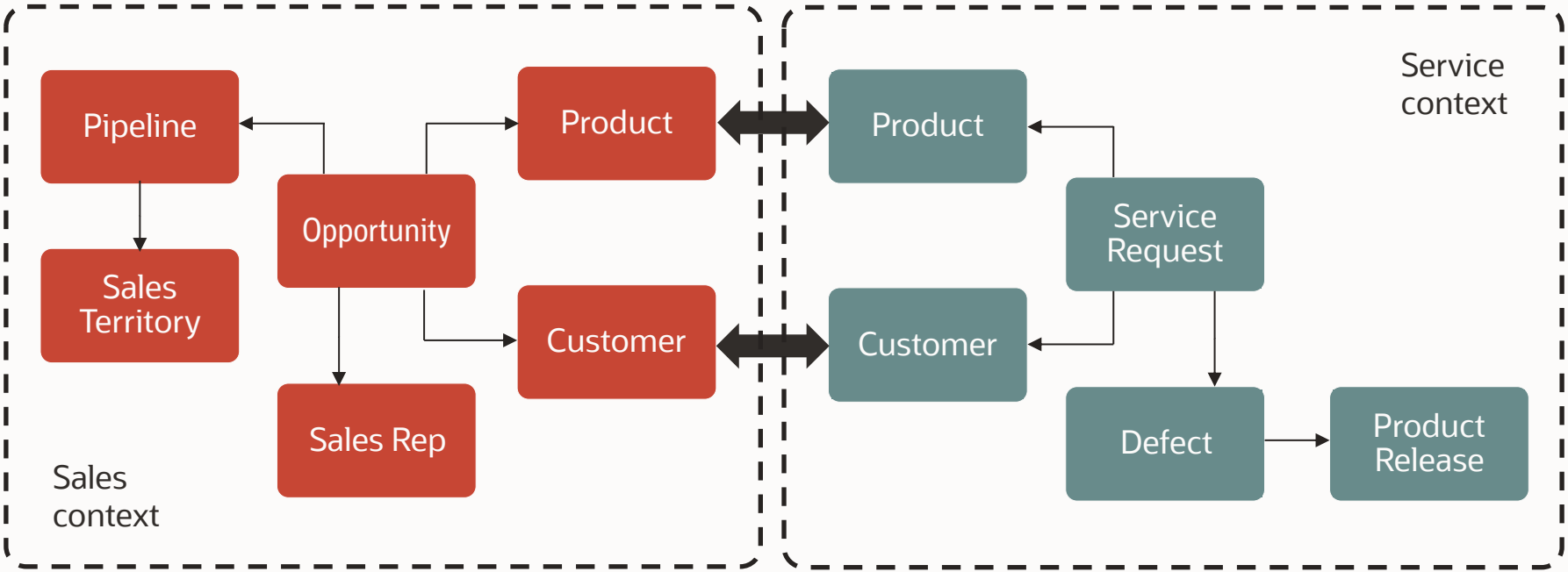
Identify candidates for microservices



Copyright © 2026, Oracle and/or its affiliates.

Bounded contexts

Community detection determines logical units



Copyright © 2026, Oracle and/or its affiliates.

Blog post and Github repo by Doug Drechsel

https://github.com/oracle/microservices-datadriven/tree/main/data-refactoring-advisor/hands-on-lab

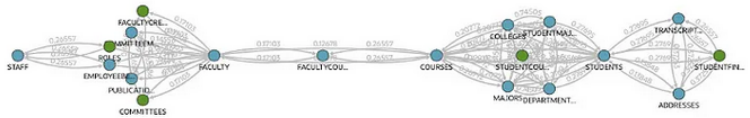
Medium

From Tables to Contexts: Reveal Domain-Driven Design Clusters in Relational Databases

Doug Does Database Follow 5 min read · Mar 11, 2025

Share

1

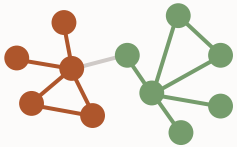


Graph of Join Activity for a University Schema

Mapping relational tables to micro service bounded contexts can be difficult. Using community detection algorithms based on a join activity graph can make this task easier.

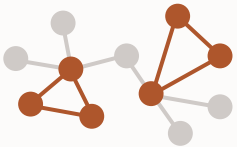
Copyright © 2026, Oracle and/or its affiliates.

Graph analytics: 80+ parallelized, in-memory algorithms out-of-the-box



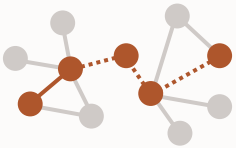
Detecting communities

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



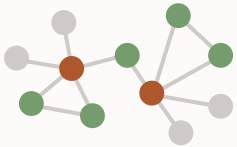
Topology analysis

Conductance, Cycle Detection, Degree Distribution, Eccentricity, K-Core, LCC, Modularity, Reachability Topological Ordering, Triangle Counting, Bipartite Check, Partition conductance



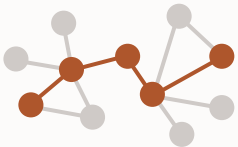
Link prediction and others

Twitter Whom-to-follow, SALSA, Adamic-Adar Index



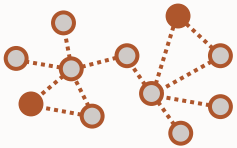
Ranking and walking

PageRank, Personalized PageRank, Degree Centrality, Closeness Centrality, Vertex Betweenness Centrality, Eigenvector Centrality, HITS, Minimum Spanning-Tree (Prim's), Breadth-First Search, Depth-First Search, Random Walk with Restart



Path-finding

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



Machine learning

DeepWalk, Supervised GraphWise, Unsupervised GraphWise, Pg2Vec, Matrix Factorization, GNNExplainer

Copyright © 2026, Oracle and/or its affiliates.