

Oracle 23c对Graph图数据库的增强

公益讲座11: 00准时开始, 请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布, 已入群客户请勿重复入群!



20-22

数据库和云讲座群



甲骨文云技术公众号



B站专家系列课程



基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

400-699-8888

* 活动最终解释权归甲骨文公司所有

Oracle 23c 对 Graph 图数据库的增强

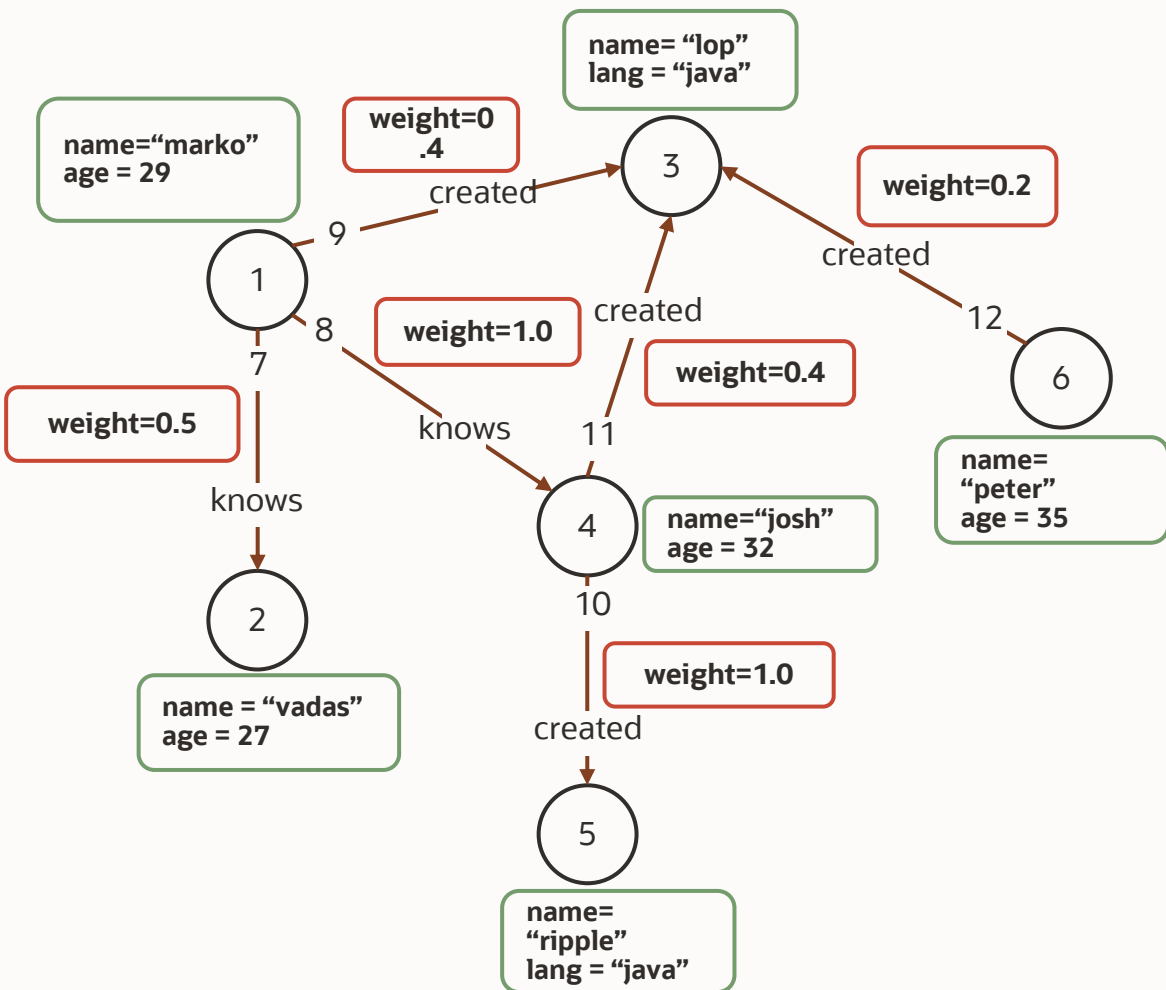
甲骨文技术公益课 - 数据库专场

2023年11月10日 11:00

线上直播

梁山

什么是图数据模型



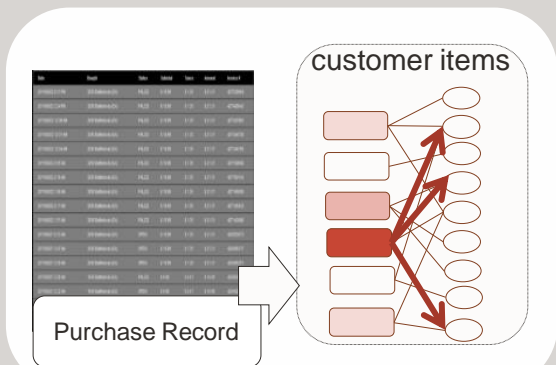
- 一组顶点 (或节点)
 - each vertex has a unique identifier.
 - each vertex has a set of in/out edges.
 - each vertex has a collection of **key-value** properties.
- 一组边(或 连线)
 - each edge has a unique identifier.
 - each edge has a head/tail vertex.
 - each edge has a label denoting type of relationship between two vertices.
 - each edge has a collection of **key-value** properties.



图分析的应用场景

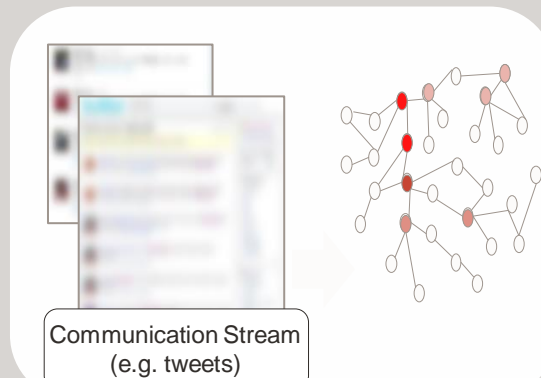
向有相似喜好的顾客推荐相应的商品

产品推荐



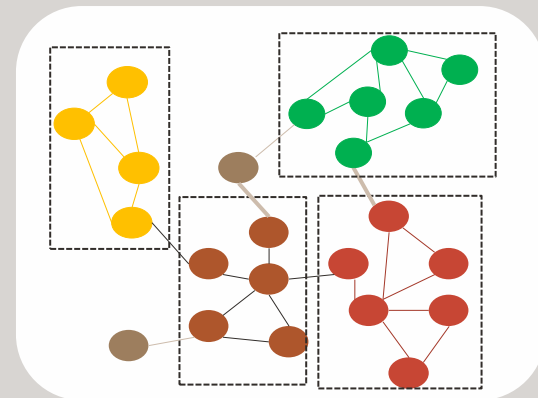
寻找某网络里的中心人物
- e.g. 市场引领者

引领者识别



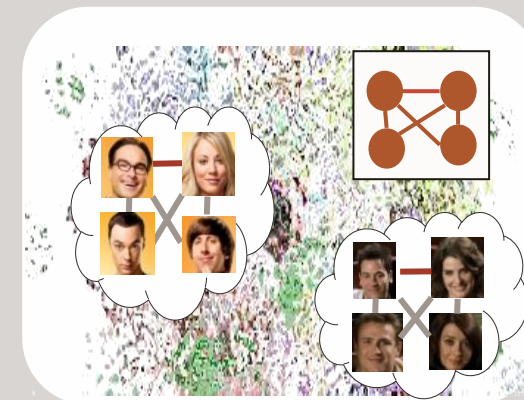
界定关系密切的人群-
e.g. 定位消费群体

社区发现



找到符合某种模式所有的个体

图形模式匹配

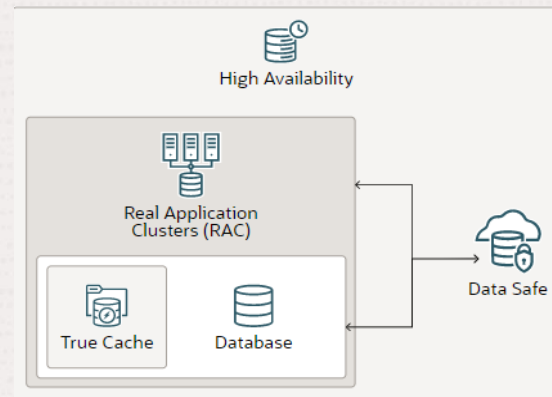


Oracle Database 23c App Simple



Key Points

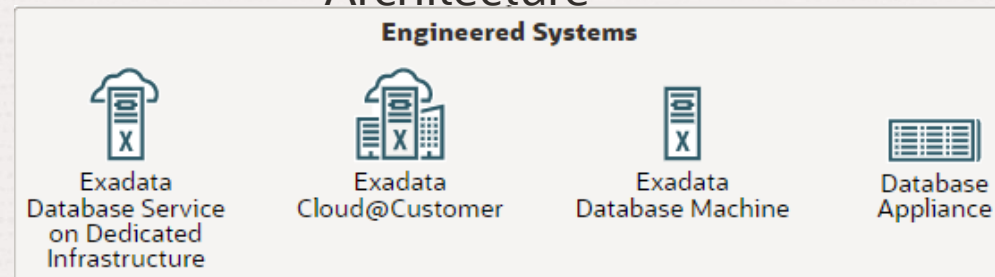
- ❑ 下一个Long Term Release
- ❑ 继续为on-premises和云端提供融合数据库创新
- ❑ 带来了众多新功能，简化任何规模或重要性的数据驱动应用程序的开发
- ❑ 升级到19c, 为未来直接升级23c做好准备



值得关注的特性

- True Cache
- JSON Relational Duality (JSON关系二元性)
- Up to 4096 Columns per Table
- Lock-free Column Value Reservations
- SQL Firewall
- ADG for PDB (DGPDB)
- In-Database Property Graph

23c Technical Architecture



Oracle Database Release 23c 对Graph 图数据的管理增强

SQL Property Graphs

- 基于SQL 方式对图数据进行管理，更加简单易用

SQL Graph Query (vs. PGQL Property Graphs)

- 更加符合SQL的语法特点，进一步降低开发难度

Aggregate Functions in SQL Graph Queries

- 在图SQL 查询中增加了聚合计算函数

Loading a Subgraph Using SQL Property Graphs and support expanding subgraph

- 按SQL 方式过滤生成子图数据

Sync graph database update data to PGX

- 数据库中的图数据的修改同步到PGX图内存分析引擎中

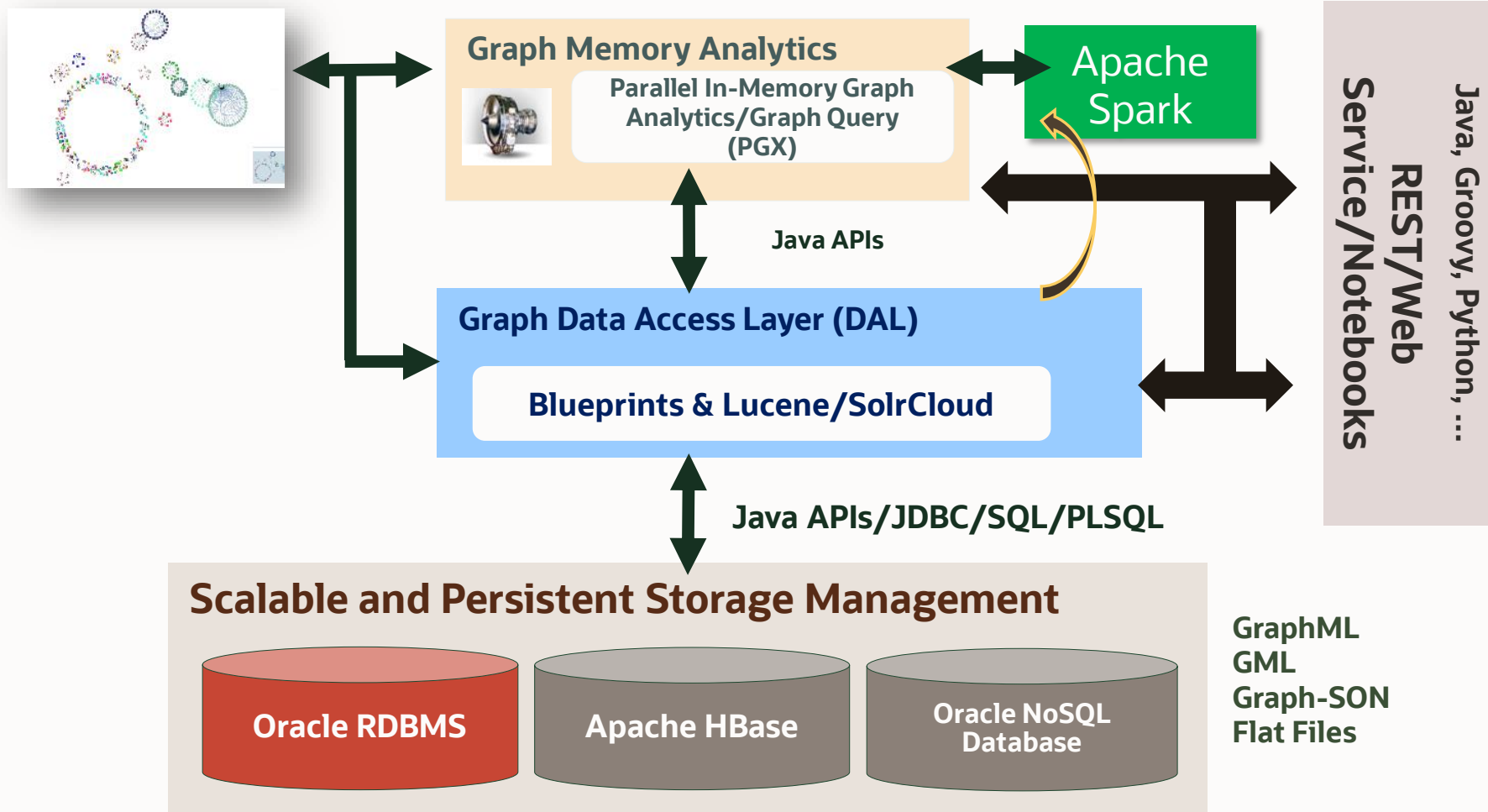
Using the Graph Visualization Application with SQL Graph Queries

- 通过SQL Graph 方式快速查询和展示图数据

Visualizing SQL Graph Queries Using the APEX Graph Visualization Plug-in

- 在APEX中嵌入SQL 图查询

Oracle Graph 架构



- In-memory graph server (PGX)
 - 内存分析
 - 大规模图分析和训练
- Property Graph Query Language
 - SQL Graph: SQL-like query language to find patterns in graphs
- 60+ pre-built graph algorithms for powerful analytics
 - Java and Python API
- 图展示工具
 - Graph Studio
 - Graph visualization



Oracle Graph 图创建：从关系表结构开始



Table	名称	数据类型	约束
BANK_ACCOUNTS	ID	NUMBER	PK 🖱️
	NAME	VARCHAR2	
	BALANCE	NUMBER	

建立图的基础表的要求

- 必须有PK

SQL Graph DDL SYNTAX:

```
CREATE PROPERTY GRAPH bank_graph
```

```
  VERTEX TABLES (
    ACCOUNTS LABEL Account PROPERTIES ( ID )
  )
```

```
  EDGE TABLES (
    BANK_TRANSFERS
```

```
    KEY(TXN_ID)
```

```
    SOURCE KEY (SRC_ACCT_ID) REFERENCES BANK_TRANSFERS
```

```
    DESTINATION KEY ( TO_ACCT_ID ) REFERENCES BANK_TRANSFERS
```

```
    LABEL Transfer PROPERTIES ( AMOUNT )
```

Table	名称	数据类型	约束
BANK_TRANSFERS	TXN_ID	NUMBER	P
	SRC_ACCT_ID	VARCHAR2	F
	DST_ACCT_ID	NUMBER	F
	DESCRIPTION	VARCHAR2	
	AMOUNT	NUMBER	

图数据

Demo: 银行交易数据



ADMIN.BANK_ACCOUNTS

列	插入	删除所选项	提交	全部撤消
数据				
	ID	NAME	BALANCE	
1	509	LANCE YEAKEL	166972.49	
2	510	GLADIS TREVEY	115705.48	
3	512	SHANICE VINTON	13380.44	
4	513	GRISELDA KUROHAI	119180.03	
5	515	KIMBRA DING	135178.67	
6	516	SCOTTY GOFFREDO	130302.93	
7	518	TIESHA CACACE	74782	
8	519	WINFRED SOHN	64659.62	
9	520	RUSSELL RIVERA	37427.08	
10	522	ANGEL SAKODA	64371.71	
11	523	ANNMARIE ESMAY	75395.04	
12	525	DARLEEN CASTEN	137771.46	
13	527	PAGE WHETZELL	53297.24	

ADMIN.BANK_TRANSFERS

列	插入	删除所选项	提交	全部撤消	
数据					
	TXN_ID	SRC_ACCT_ID	DST_ACCT_ID	DESCRIPTION	AMOUNT
1	486	268	891	transfer	1185
2	487	269	775	transfer	2475
3	488	269	519	transfer	2781
4	489	269	759	transfer	6600
5	490	269	52	transfer	1782
6	491	269	980	transfer	5503
7	492	270	495	transfer	9638
8	493	270	771	transfer	7353
9	494	270	469	transfer	2162
10	495	270	407	transfer	8054
11	496	270	724	transfer	6061
12	497	271	419	transfer	1426
13	498	271	593	transfer	5519



Oracle Graph 图模式: Internal表结构

建立图后系统

Table	ADMIN.BANK_GRAPH_PROPERTY\$
BANK	列 <input type="checkbox"/> 插入 <input type="checkbox"/> 删除所选项 <input checked="" type="checkbox"/> 提交 <input type="checkbox"/> 全部撤消
BANK	数据
BANK	约束条件
BANK	授权
BANK	统计信息
BANK	触发器
BANK	相关性
BANK	详细信息

	PROPERTY_NAME	ET_NAME	ET_TYPE	LABEL_NAME	COLUMN_NAME
1	BALANCE	BANK_ACCOUNTS	VERTEX	BANK_ACCOUNTS	BALANCE
2	ID	BANK_ACCOUNTS	VERTEX	BANK_ACCOUNTS	ID
3	NAME	BANK_ACCOUNTS	VERTEX	BANK_ACCOUNTS	NAME
4	AMOUNT	BANK_TRANSFERS	EDGE	BANK_TRANSFERS	AMOUNT
5	DESCRIPTION	BANK_TRANSFERS	EDGE	BANK_TRANSFERS	DESCRIPTION
6	DST_ACCT_ID	BANK_TRANSFERS	EDGE	BANK_TRANSFERS	DST_ACCT_ID
7	SRC_ACCT_ID	BANK_TRANSFERS	EDGE	BANK_TRANSFERS	SRC_ACCT_ID
8	TXN_ID	BANK_TRANSFERS	EDGE	BANK_TRANSFERS	TXN_ID



图数据的查询方式

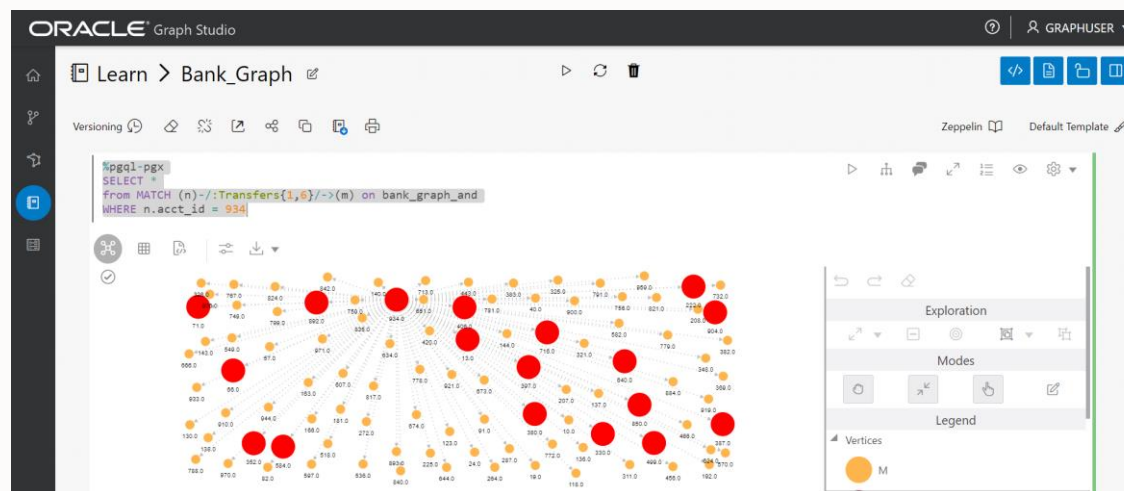


Java and Python API

- pagerank (<graph>)
- betweennessCentrality (<graph>)
- WhomToFollow (<vertex>)

Use analytics results in SQL Graph queries

- ```
SELECT a.acct_id, a.pagerank
FROM MATCH(a)
ORDER BY a.pagerank DESC
ON graph
```



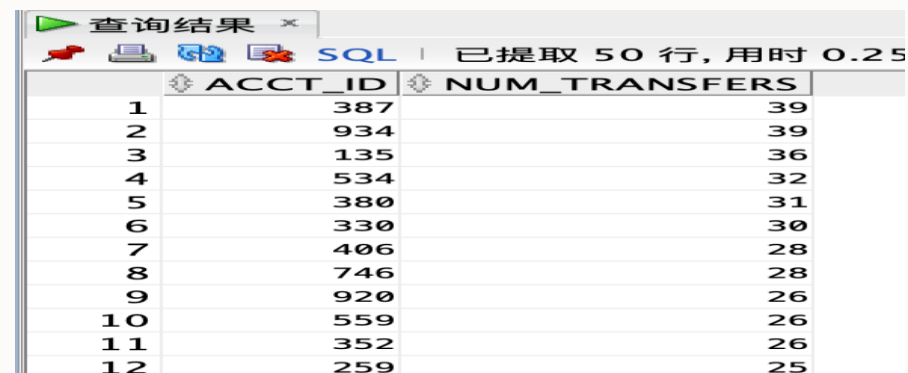
# 查询Graph 图数据 - 分析异常交易

Oracle 通过 SQL GRAPH Query 查询语句快速、简单的方式完成对图数据的分析

- SQL-like query language with the ability to specify graph patterns

## 1、找出转入次数最多的前10名账号

```
SELECT acct_id, COUNT(1) AS Num_Transfers
FROM graph_table (BANK_GRAPH
MATCH (src) - [IS BANK_TRANSFERS] -> (dst)
COLUMNS (dst.id AS acct_id)
) GROUP BY acct_id ORDER BY Num_Transfers DESC
FETCH FIRST 100 ROWS ONLY;
```



|    | ACCT_ID | NUM_TRANSFERS |
|----|---------|---------------|
| 1  | 387     | 39            |
| 2  | 934     | 39            |
| 3  | 135     | 36            |
| 4  | 534     | 32            |
| 5  | 380     | 31            |
| 6  | 330     | 30            |
| 7  | 406     | 28            |
| 8  | 746     | 28            |
| 9  | 920     | 26            |
| 10 | 559     | 26            |
| 11 | 352     | 26            |
| 12 | 259     | 25            |

## 2、经过3-5次转账到本账号的账号，按次数降序，取10行

```
SELECT DISTINCT(account_id), COUNT(1) AS
Num_Cycles
FROM graph_table(BANK_GRAPH
MATCH (v1)-[IS BANK_TRANSFERS]->{3, 5}(v1)
COLUMNS (v1.id AS account_id)
) GROUP BY account_id ORDER BY Num_Cycles
DESC FETCH FIRST 10 ROWS ONLY;
```



|    | ACCOUNT_ID | NUM_CYCLES |
|----|------------|------------|
| 1  | 135        | 37         |
| 2  | 387        | 34         |
| 3  | 934        | 30         |
| 4  | 640        | 28         |
| 5  | 13         | 27         |
| 6  | 458        | 27         |
| 7  | 559        | 25         |
| 8  | 352        | 23         |
| 9  | 406        | 23         |
| 10 | 499        | 22         |

# 查询 Graph 图数据 - 分析异常交易

1、找出作为两级转账的前10名账号：即 A转账B， B转账到C

```
SELECT acct_id, COUNT(1) AS Num_In_Middle
FROM graph_table (BANK_GRAPH
MATCH (src) - [IS BANK_TRANSFERS] -> (via) -
[IS BANK_TRANSFERS] -> (dst)
COLUMNS (via.id AS acct_id)
) GROUP BY acct_id ORDER BY Num_In_Middle
DESC FETCH FIRST 10 ROWS ONLY;
```

2、在1、2或3跳中从帐户387收到转账的帐户: 387 转账给A (1级), A转账给B (2级), B转账给C (3级), 有多少这样的A、B、C账号

```
SELECT account_id1, account_id2
FROM graph_table(BANK_GRAPH
MATCH (v1)-[IS BANK_TRANSFERS]->{1,3}(v2)
WHERE v1.id = 387
COLUMNS (v1.id AS account_id1, v2.id AS
account_id2)
);
```

查询结果 x

提取的所有行: 10, 用时 0.267 秒

|    | ACCT_ID | NUM_IN_MIDDLE |
|----|---------|---------------|
| 1  | 387     | 195           |
| 2  | 934     | 195           |
| 3  | 135     | 180           |
| 4  | 534     | 160           |
| 5  | 380     | 155           |
| 6  | 330     | 150           |
| 7  | 406     | 140           |
| 8  | 746     | 140           |
| 9  | 920     | 130           |
| 10 | 352     | 130           |

查询结果 x

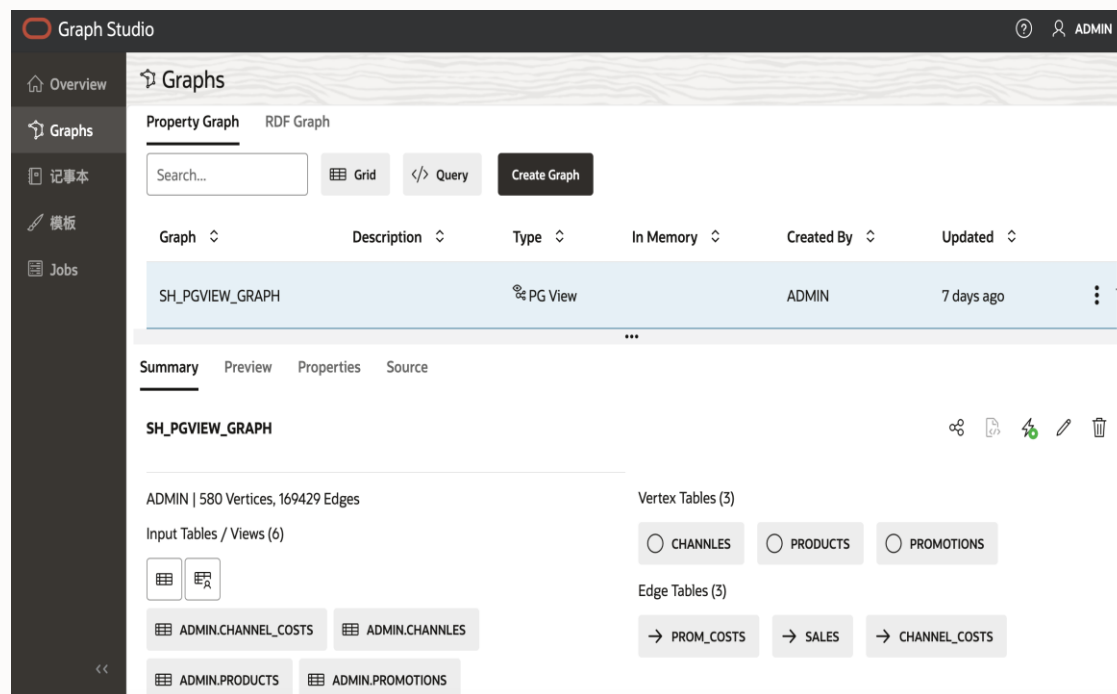
已提取 50 行, 用时 0.507 秒

|    | ACCOUNT_ID1 | ACCOUNT_ID2 |
|----|-------------|-------------|
| 1  | 387         | 577         |
| 2  | 387         | 867         |
| 3  | 387         | 998         |
| 4  | 387         | 188         |
| 5  | 387         | 374         |
| 6  | 387         | 926         |
| 7  | 387         | 63          |
| 8  | 387         | 183         |
| 9  | 387         | 463         |
| 10 | 387         | 581         |
| 11 | 387         | 678         |

# 图开发工具 Graph Studio

Graph Studio 提供简单的快速创建图模型的工具集合，使用户通过界面的方式管理图数据，如图的创建、图查询、图分析和图展现等各个操作图等环节

- 简单的图创建工作流引擎
  - 基于现有表模式转换到图模式
  - 图属性的管理
  - 图标签管理
  - 图结构预览
- 简单易用的图数据操作界面
  - 基于实体关系表的方式访问图数据
- 图数据库展示和Apex协同
- 加载图数据库到PGX 内存引擎中
- 图分析引擎的环境管理和配置



# 基于关系模型的图数据分析和开发

-根据PangRank 算法，找出连接数最多的账号

%python graph analysis

Vertex

```
%pgql-pgx
/* List accounts in descending order of pagerank values*/
SELECT a.acct_id, a.pagerank AS pagerank
FROM MATCH (a) ON bank_graph
ORDER BY PageRank DESC
LIMIT 10
```

输入以搜索

| ACCT_ID | PAGERANK              |
|---------|-----------------------|
| 387     | 0.007302836252205922  |
| 406     | 0.006734430614559079  |
| 135     | 0.006725965475577351  |
| 934     | 0.006641340764834485  |
| 397     | 0.0057016075312134595 |





# 基于关系模型的图数据分析和开发

-开发图数据就像普通的关系数据库表操作一样

The screenshot shows a database interface with a 'Query' tab and a 'Create Graph' button. Below the button is a table with columns: Type, In Memory, Created By, and Updated. The table contains one entry: PG View, in memory, ADMIN, 7 days ago. At the bottom, there are sections for 'Vertex Tables (3)' and 'Edge Tables (3)'. The 'Vertex Tables' section includes buttons for CHANNLES, PRODUCTS, and PROMOTIONS. The 'Edge Tables' section includes buttons for PROM\_COSTS, SALES, and CHANNEL\_COSTS. A red box highlights these two sections.

| Type    | In Memory | Created By | Updated    |
|---------|-----------|------------|------------|
| PG View | in memory | ADMIN      | 7 days ago |

Vertex Tables (3)

- CHANNLES
- PRODUCTS
- PROMOTIONS

Edge Tables (3)

- PROM\_COSTS
- SALES
- CHANNEL\_COSTS

The screenshot shows a database interface with a 'Query' tab and a 'Create Graph' button. Below the button is a table with columns: Type, In Memory, Created By, and Updated. The table contains one entry: PG View, in memory, ADMIN, 7 days ago. At the bottom, there are sections for 'Vertex Tables (3)' and 'Edge Tables (3)'. The 'Vertex Tables' section includes buttons for CHANNLES, PRODUCTS, and PROMOTIONS. The 'Edge Tables' section includes buttons for PROM\_COSTS, SALES, and CHANNEL\_COSTS. A red box highlights these two sections.

```
match (s:PRODUCTS) - [e:PRD_CHANNEL_COSTS] -> (c:CHANNLES) on SH_PGVIEW_GRAPH
where c.id > 20
```

751 页 | 1 2 3 4 ... 751 最后一页



# 基于关系模型的图数据分析和开发



```
%pgql-pgx

SELECT LABEL(v) as Vertex_Label, COUNT(v) as Number_of_Vertices
FROM MATCH (v) ON SH_PGVIEW_GRAPH
GROUP BY Vertex_Label
```

输入以搜索

| VERTEX_LABEL | NUMBER_OF_VERTICES |
|--------------|--------------------|
| PRODUCTS     | 72                 |
| PROMOTIONS   | 503                |
| CHANNLES     | 5                  |

页 1 , 共 1 页 (第 1-3 项, 共 3 项) 加载更多

```
%pgql-pgx

SELECT LABEL(e) as Edge_Label, COUNT(e) as Number_of_Edges
FROM MATCH ()-[e]->() ON SH_PGVIEW_GRAPH
GROUP BY Edge_Label
```

输入以搜索

| EDGE_LABEL        | NUMBER_OF_EDGES |
|-------------------|-----------------|
| PRD_CHANNEL_COSTS | 82112           |
| PRO_CNEL_SALES    | 5205            |
| PROM_PRD_COSTS    | 82112           |

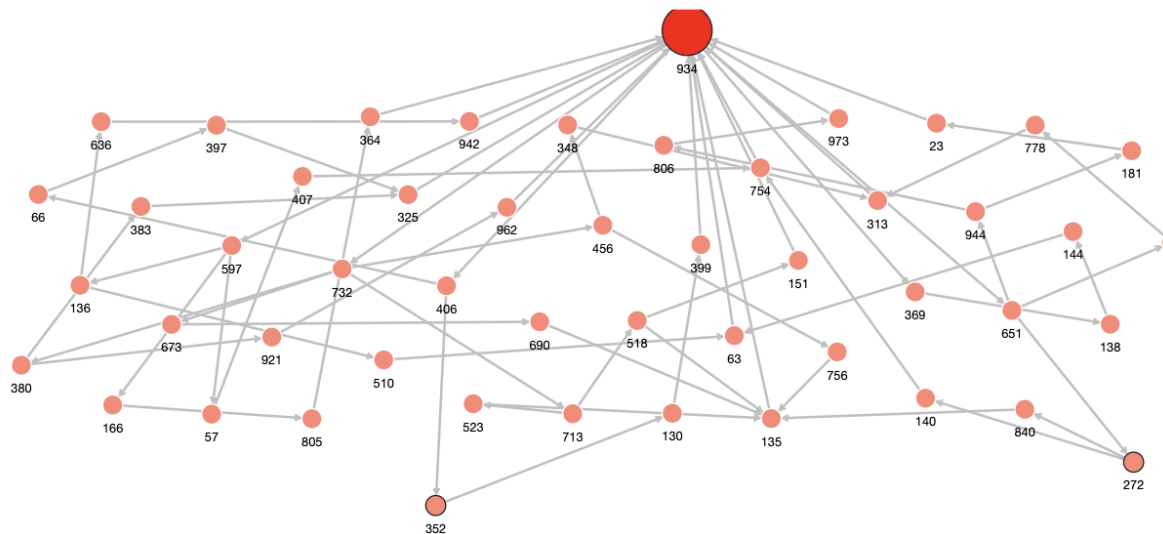
页 1 , 共 1 页 (第 1-3 项, 共 3 项) 加载更多



# 通过 Graph Studio 查询图数据 - 分析异常交易

找出参与了任何源自和终止于账户 934 的4次循环转账

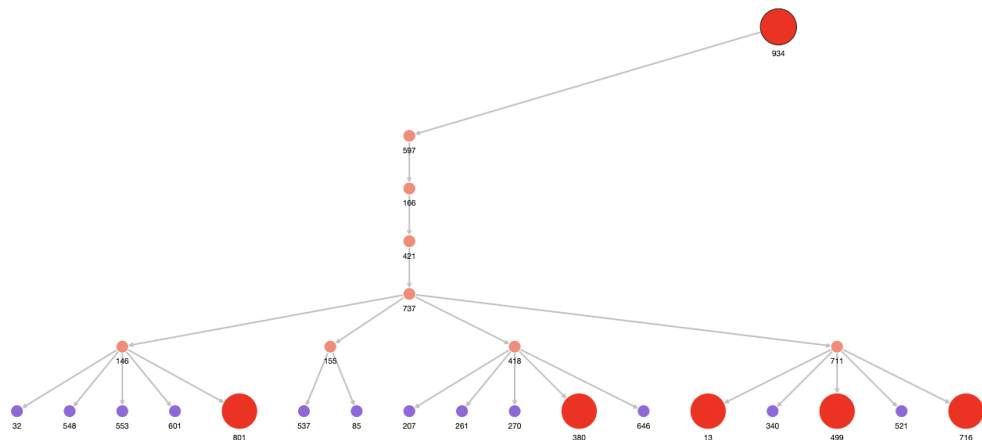
```
%pgql-pgx
/* Check if there are any circular payment chains of length 5 from acct 934 */
SELECT v,e,v2
FROM MATCH ALL (a)-[:TRANSFERS]->{5}(b) ON bank_graph ONE ROW PER STEP (v,e,v2)
WHERE a.acct_id=934 AND id(a) = id(b)
LIMIT 100
```



# 通过 Graph Studio 查询图数据 - 分析异常交易

找出账户 934 的6跳支付链

```
%pgql-pgx
/* Add highlights to symbolize account nodes by PageRank values. This shows that 934 and highlights accounts with
Choose the hierarchical view. */
SELECT v,e,v2
FROM MATCH ANY (n)-[:Transfers]->{6}(m) ON bank_graph ONE ROW PER STEP (v,e,v2)
WHERE n.acct_id = 934
LIMIT 100
```



# 通过 Graph Studio 查询图数据 - 分析异常交易

检查两个账号是否存在转账关系

```
%pgql-pgx
/* Check the shortest path between account 934 and account 387 */
SELECT v,e,v2
FROM MATCH SHORTEST (a)-[:TRANSFERS]->+(b) ON bank_graph ONE ROW PER STEP (v,e,v2)
WHERE a.acct_id=934 AND b.acct_id=387
```

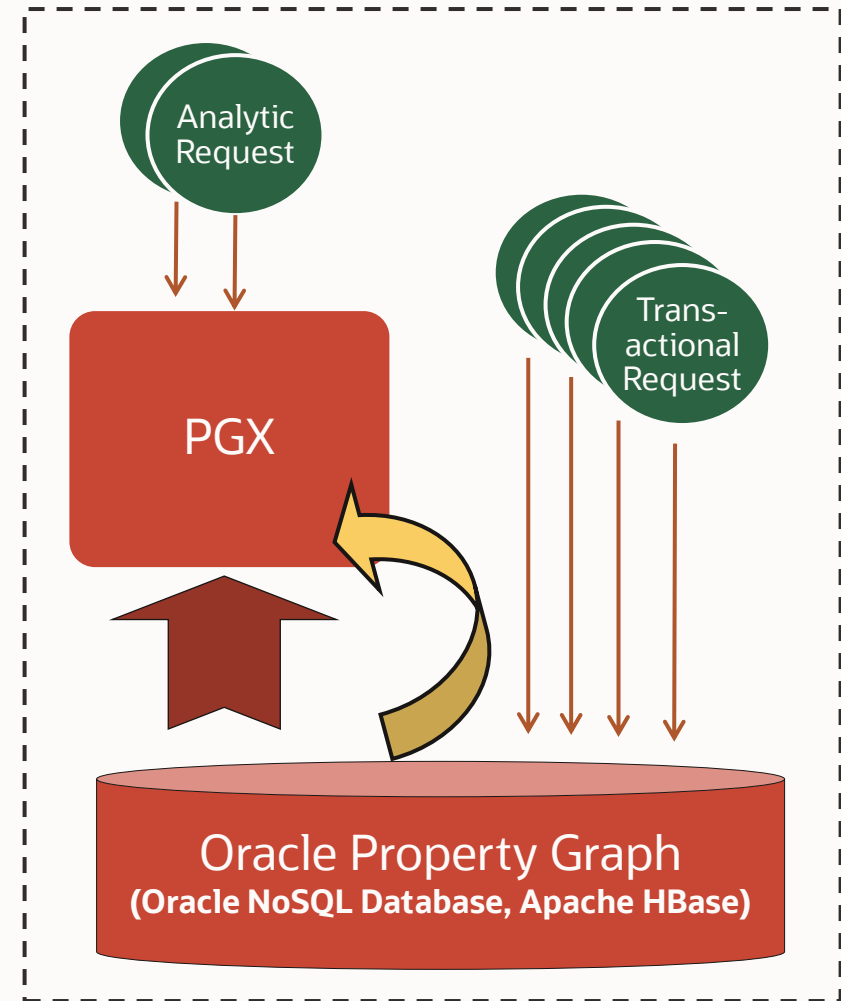


# Oracle Graph特点之一：基于内存的并行图计算 - PGX

基于SQL Graph Table 图加载到内存中计算

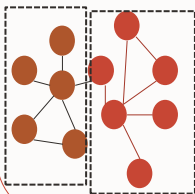
```
PgxGraph graph = session.readSubgraph()
 .fromPgSql("STUDENT_NETWORK")
 .queryPgql("MATCH (v1 IS Person)-[e I:
 .queryPgql("MATCH (v:Person) WHERE id
 .load("student_subgraph");
```

```
= session.readGraphByName("STUDENT_NETWORK")
on.optimizeFor(GraphOptimizedFor.READ);
```



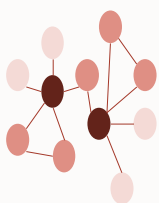
# Oracle Graph特点之二: 60+ Graph 图算法支持

## Detecting Components and Communities 监测群体和社区



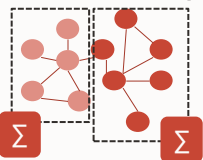
- Tarjan's, Kosaraju's,
- Weakly Connected Components, Label Propagation (w/ variants),
- Spasification

## Ranking and Walking 排名



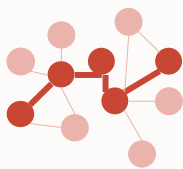
- Pagerank, Personalized Pagerank,
- Betweenness Centrality (w/ variants),
- Closeness Centrality, Degree Centrality,
- Eigenvector Centrality, HITS,
- Random walking and sampling (w/ variants)

## Evaluating Community Structures 社区结构分析



- Conductance, Modularity
- Clustering Coefficient (Triangle Counting)

## Path-Finding 路径查找



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

## Link Prediction 链路预测



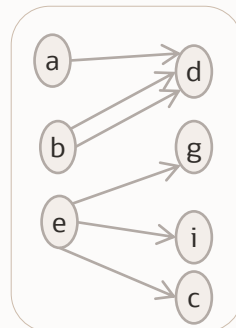
- WTF (Who to follow)
- Minimum Spanning-Tree,
- Matrix Factorization

## Machine learning



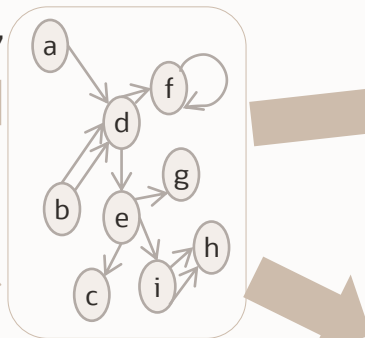
- DeepWalk
- Supervised GraphWise
- Pg2Vec

The original graph



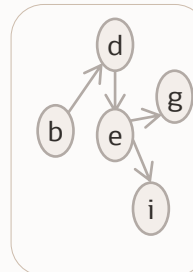
二分图

Left Set: "a,b,e"



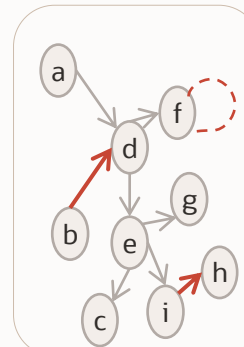
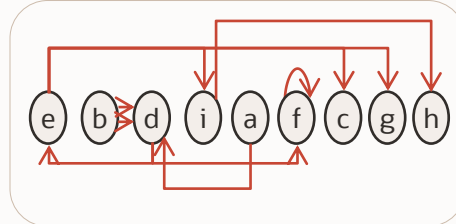
无方向图

Filter-Expression



过滤子图

按度级排序 (Renumbering)



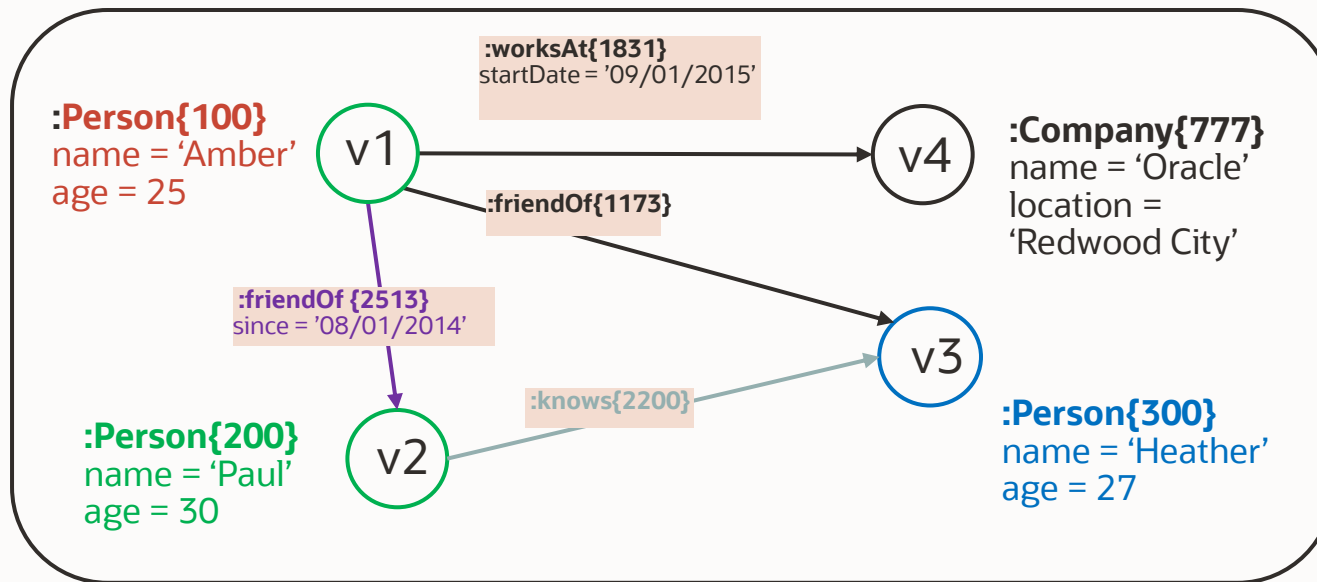
简化 Graph



# Oracle Graph特点之三：SQL Graph操作的支持

查询: 谁认识Amber的朋友

```
SELECT v3.name, v3.age
FROM myGraph
WHERE
 (v1 WITH name = 'Amber') -[:friendOf]-> (v2) -[:knows]-> (v3)
```



property graph  
'myGraph'

query

solutions

| v3.name   | v3.age |
|-----------|--------|
| 'Heather' | 27     |





# Recursive Path Query

## PGQL vs. SQL

图格式查询 vs 普通关系查询代码量区别

### 图查询SQL

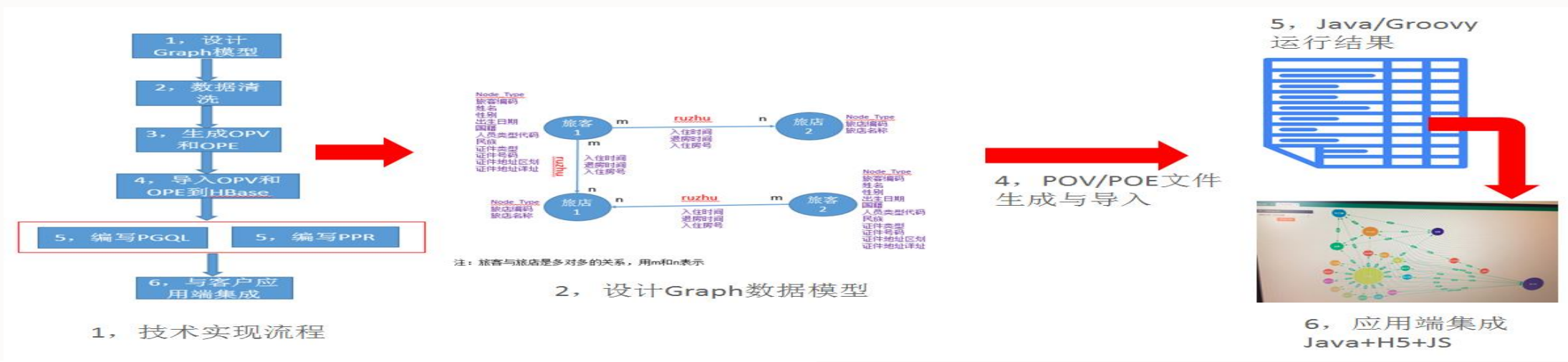
```
PATH connects_to := (from) <- (connector) ->
(to)
SELECT y.name
WHERE (x:Device) -/:connects_to*/-> (y:Device),
 x.name = 'Regulator, HVMV_Sub_RegB'),
 x != y
```

VS

## 常规关系SQL

```
WITH temp(device_id, device_name) AS (
 -- Anchor member:
 SELECT device_id, name
 FROM Devices
 WHERE name = 'Regulator, HVMV_Sub_RegB'
 UNION ALL
 -- Recursive member:
 SELECT Devices.device_id, Devices.name
 FROM temp, Devices, Connections conn1,
 Connections conn2, Connectors
 WHERE temp.device_id = conn1.to_device_id
 AND conn1.from_connector_id =
Connectors.connector_id
 AND Connectors.connector_id =
conn2.from_connector_id
 AND conn2.to_device_id =
Devices.device_id
 AND temp.device_id != Devices.device_id)
CYCLE device_id SET cycle TO 1 DEFAULT 0
SELECT DISTINCT device_name
FROM temp
WHERE cycle = 0
 AND device_name != 'Regulator,
HVMV_Sub_RegB'
```

# 案例：某公安系统利用Oracle图数据库分析犯罪嫌疑人的行为

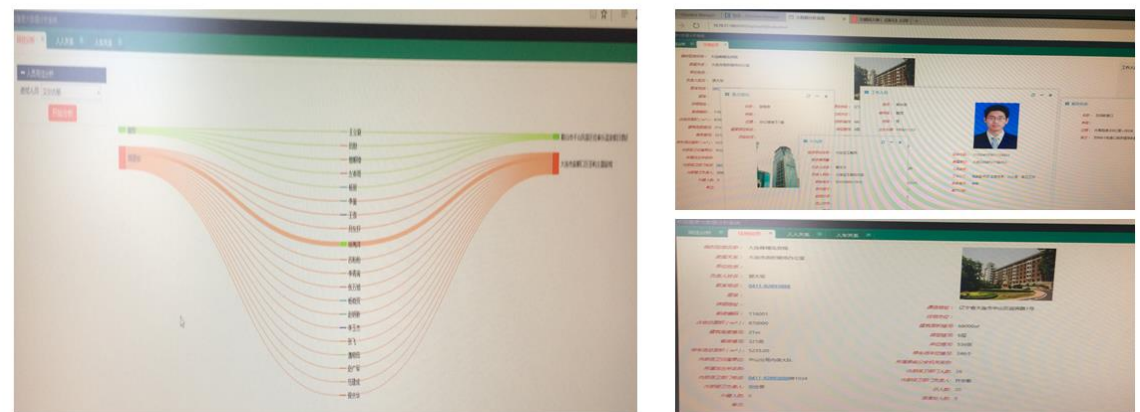


## 同住分析

指定重点人，在一个时间段内（6个小时或者12个小时），找到与这个人同住一个酒店或者同房一个酒店的同伙，层层展开随着关系链条。此场景使用PGQL实现。



## 人人关系

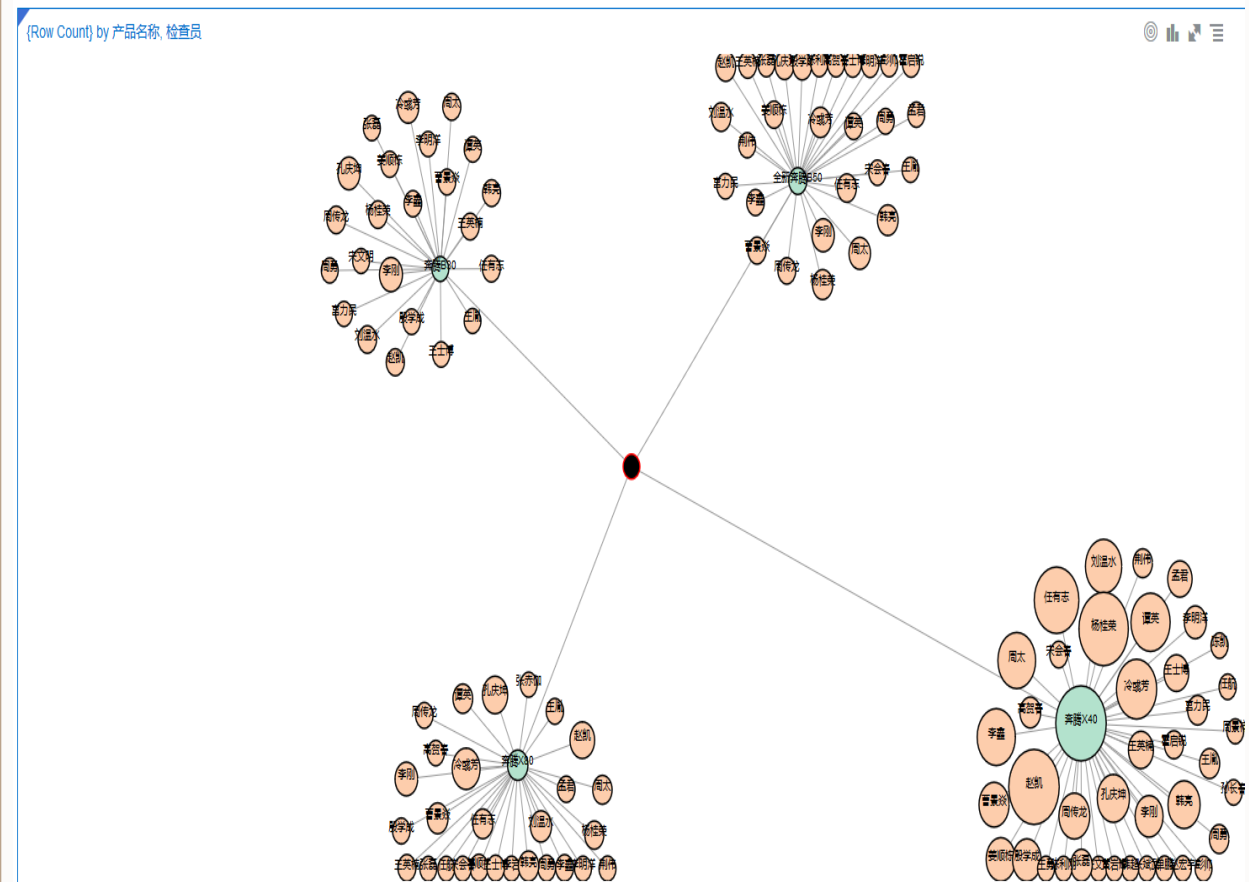
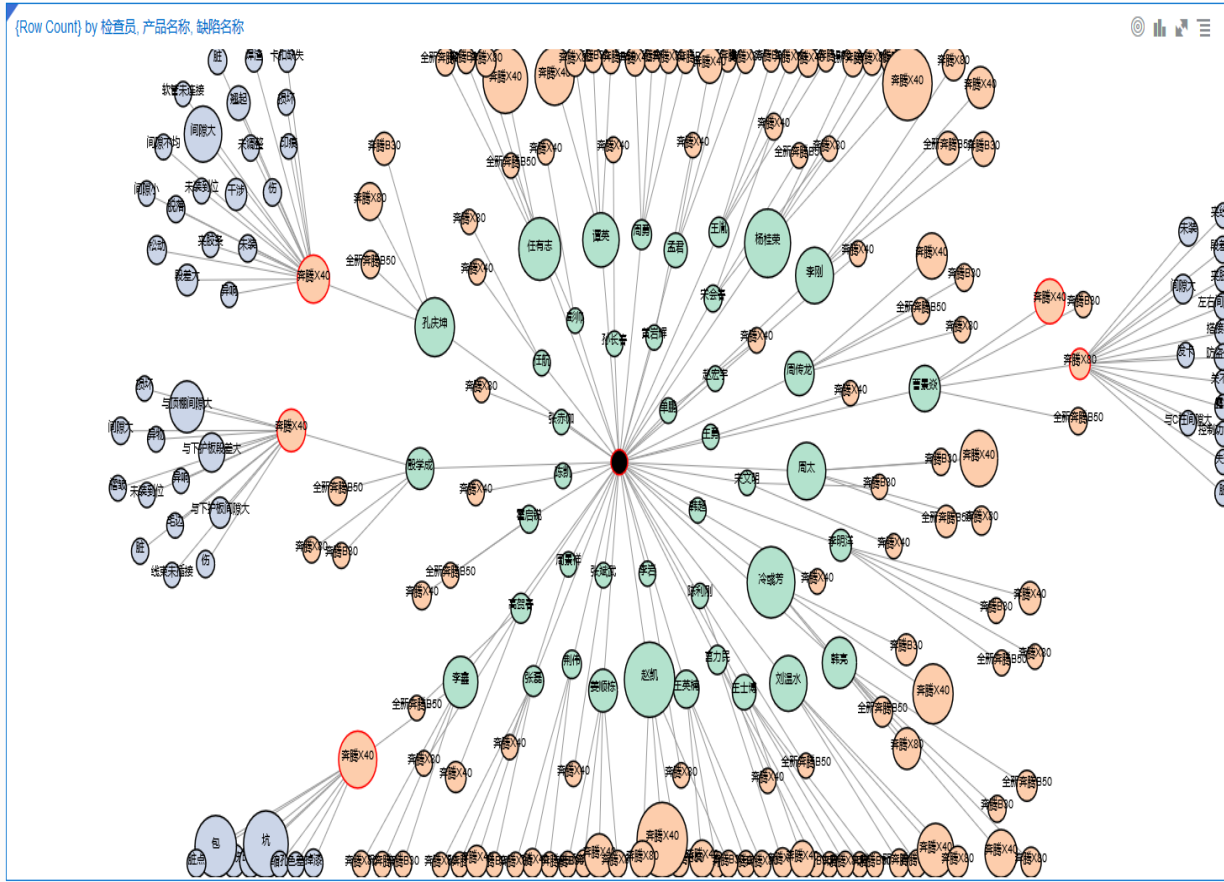


指定一个犯罪嫌疑人，找到与此犯罪嫌疑人相同特征的犯罪嫌疑人列表或相关酒店。点击可查看嫌疑人具体信息或酒店具体信息。此场景使用personalizedPageRank算法（简称PPR算法）实现，线的粗细表征关系程度。

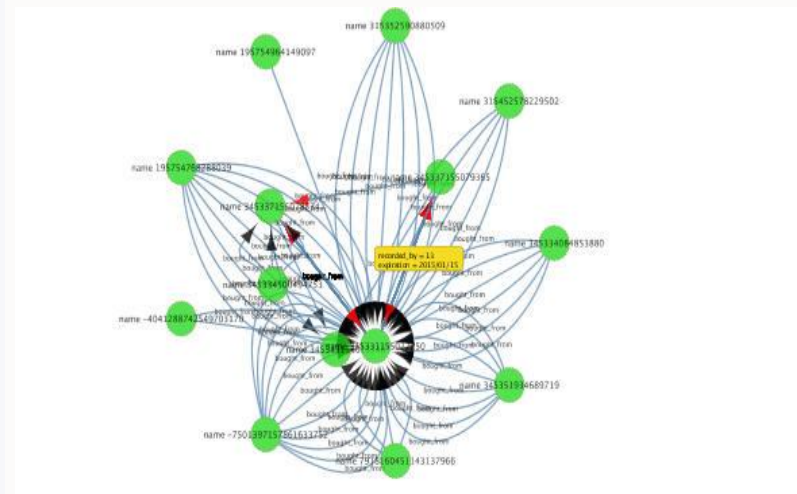
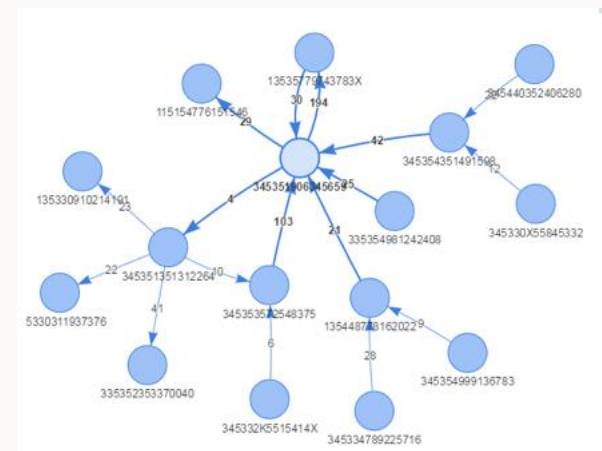
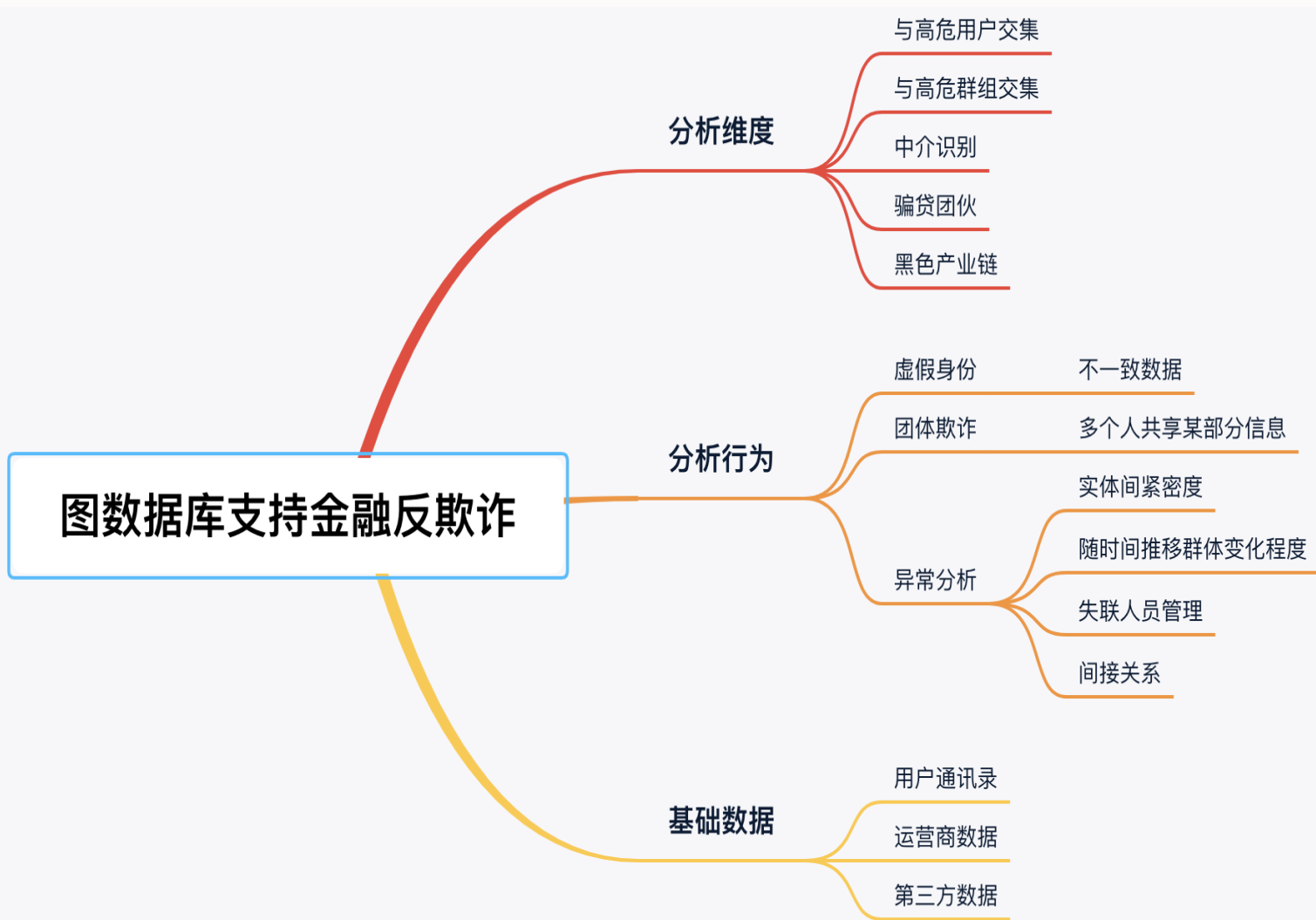


# 案例：某车企利用Oracle图数据库分析发现生产质量和流程之间的关系，优化作业调动

利用图关系计算，快速分析制造业质量问题，比如 工人、车、质量之间的关系



# 案例：某金融机构利用利用Oracle图数据库分析金融欺诈监测





# 基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

**400-699-8888**

\* 活动最终解释权归甲骨文公司所有

释放您繁重的运维工作

# Oracle自动化运维实践

数据库和云系列公益讲座



朱国璋

- 资深解决方案工程师
- 曾服务金融，制造行业ERP应用，数仓项目实施
- 专注于企业数据库领域的技术咨询和解决方案

## 内容简介

EMCC（企业管理器）提供了业界最完整的云管理方案，它为测试、部署、运营、监控、诊断和问题解决等提供了统一、集成的界面，使得IT运维从被动式反应转型到主动和预防式管理上来。

- 使用EM实现全面监控，降低运维复杂度
- 使用EM快速定位故障和性能瓶颈



Zoom直播

直播时间：11月17日 11:00 - 12:00

扫描二维码进入直播

Zoom ID: 957 9669 6723

密码: 20212023



微信扫一扫预约



数据库和云讲座群

20-22



甲骨文云技术公众号



技术专家1V1深入交流

