ORACLE

AUTONOMOUS DATABASE

LEARNING LOUNGE

Graph RAG: Bring the Power of Graphs to Generative AI

Autonomous Database Learning Lounge

Hosted by Marcos Arancibia

Autonomous Database Product Management

Agenda

Melli Annamalai

Topics

- Generative AI depends on LLMs that are trained on available data, which can be overly general or outdated resulting in errors and answers that are not relevant.
- Retrieval augmented generation (RAG) addresses these issues by enabling LLMs to be combined with private business data or other domain specific data. Graph RAG can enhance RAG by including information derived from relationships in data.
- Learn more about graphs, graph RAG and the power of combining graphs with generative AI. We will demonstrate how easy graph RAG is with graph and vector technologies in Autonomous Database.

Q&A

- **Product Managers will answer any questions**

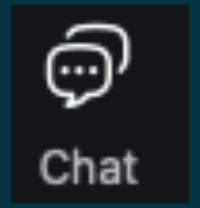
Before we begin...

This session is for you !!!

Ask your questions using **Q&A**

Product Managers are monitoring your questions

We will share links in **Chat**



The recording will be made available in a few days at
oracle.com/goto/adb-learning-lounge

Important links to bookmark

Links to get you started and to keep up to date with Autonomous Database

1. New Get Started page:
   oracle.com/autonomous-database/get-started/

2. Join us:
   LinkedIn bit.ly/adb-linkedin-grp
   @AutonomousDW

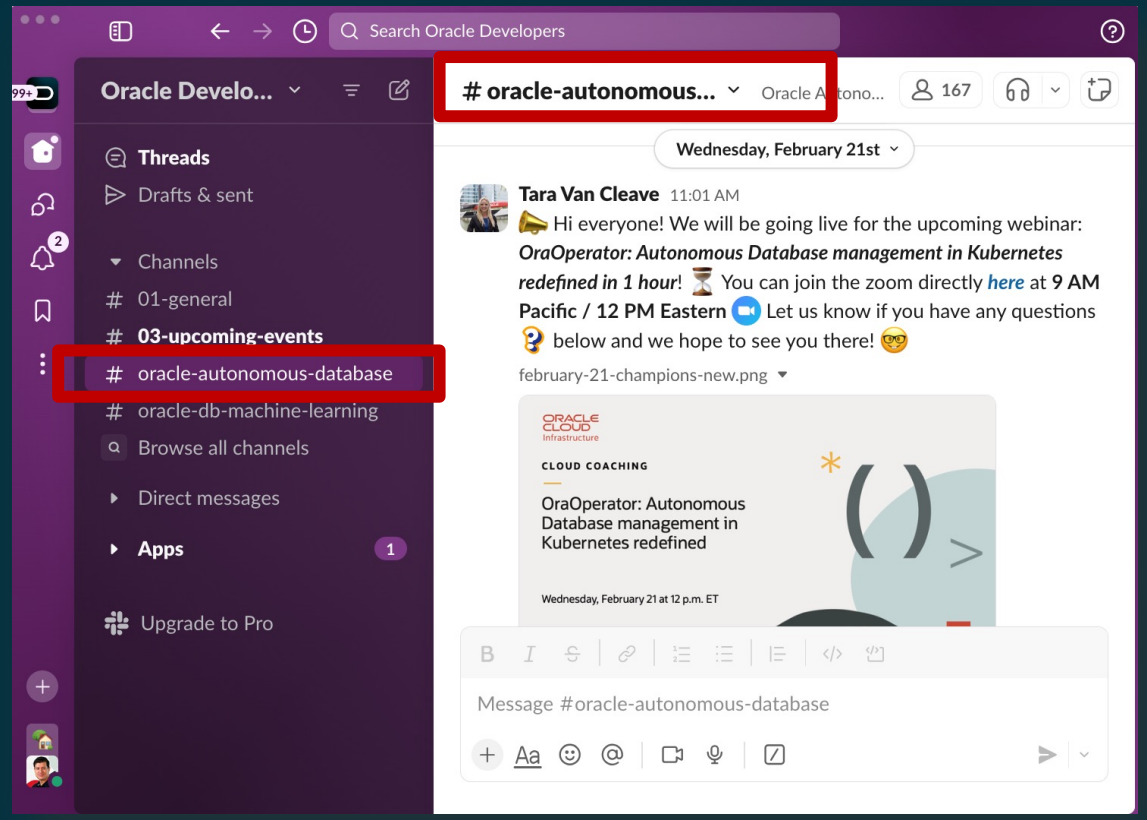
3. Got a question?
   We are on stackoverflow
   bit.ly/adb-stackoverflow

Join us on Developers Slack
(search #oracle-autonomous-database)
bit.ly/odevrel_slack (odevrel_slack)

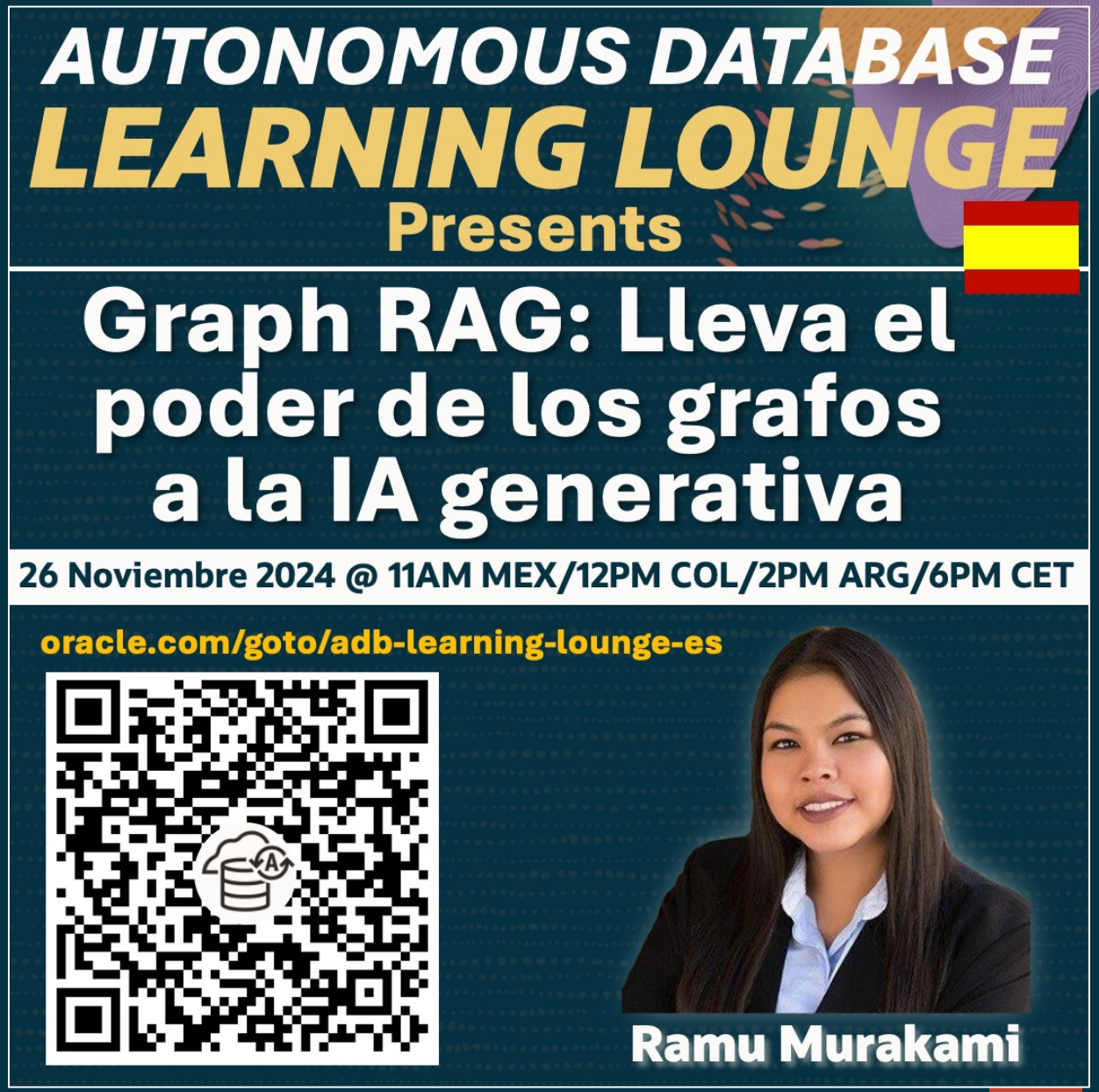
Join our External Slack

STEP 1: bit.ly/odevrel_slack (odevrel_slack)

STEP 2: search for #oracle-autonomous-database at the top and click on the Channel



Upcoming Sessions



Upcoming Sessions

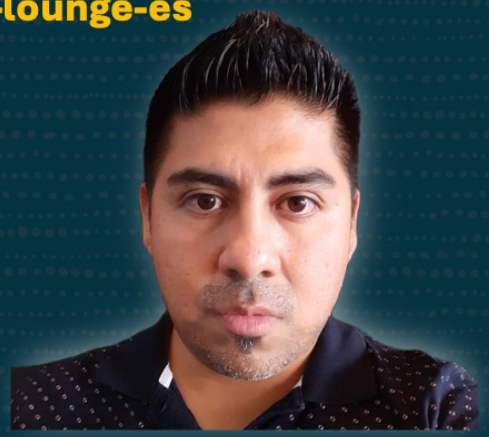
Upcoming Sessions

AUTONOMOUS DATABASE
LEARNING LOUNGE
en Español presenta

Migración a ADB Parte II: Migración sencilla desde versiones de bases de datos anteriores con DMS

3 Diciembre 2024 @ 11AM MEX/12PM COL/2PM ARG/6PM CET

oracle.com/goto/adb-learning-lounge-es



Jorge Martinez

Speaker

Melli Annamalai

ORACLE

Graph RAG: Bring the Power of Graphs to Generative AI

Melli Annamalai
Distinguished Product Manager
November 21, 2024

In collaboration with Eduard Cuba, Oracle Labs

Agenda

Quick overview of Graph

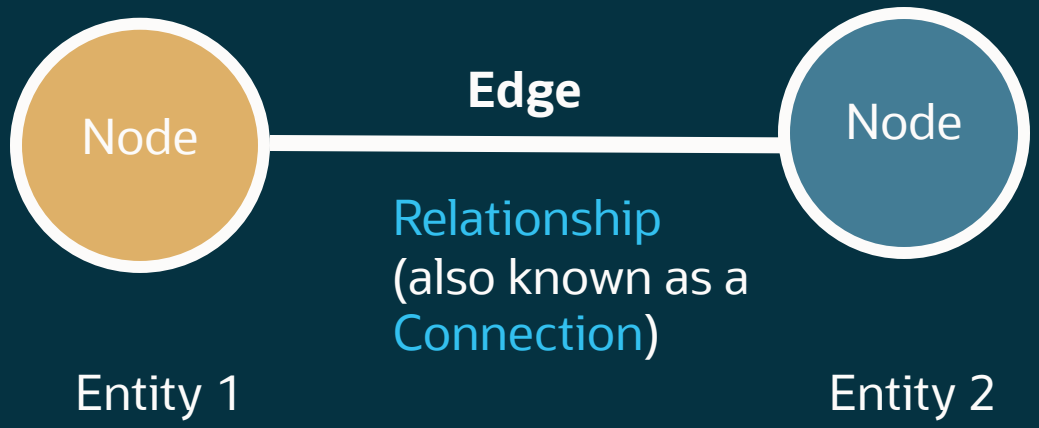
Graph RAG with three examples

What is a Graph?

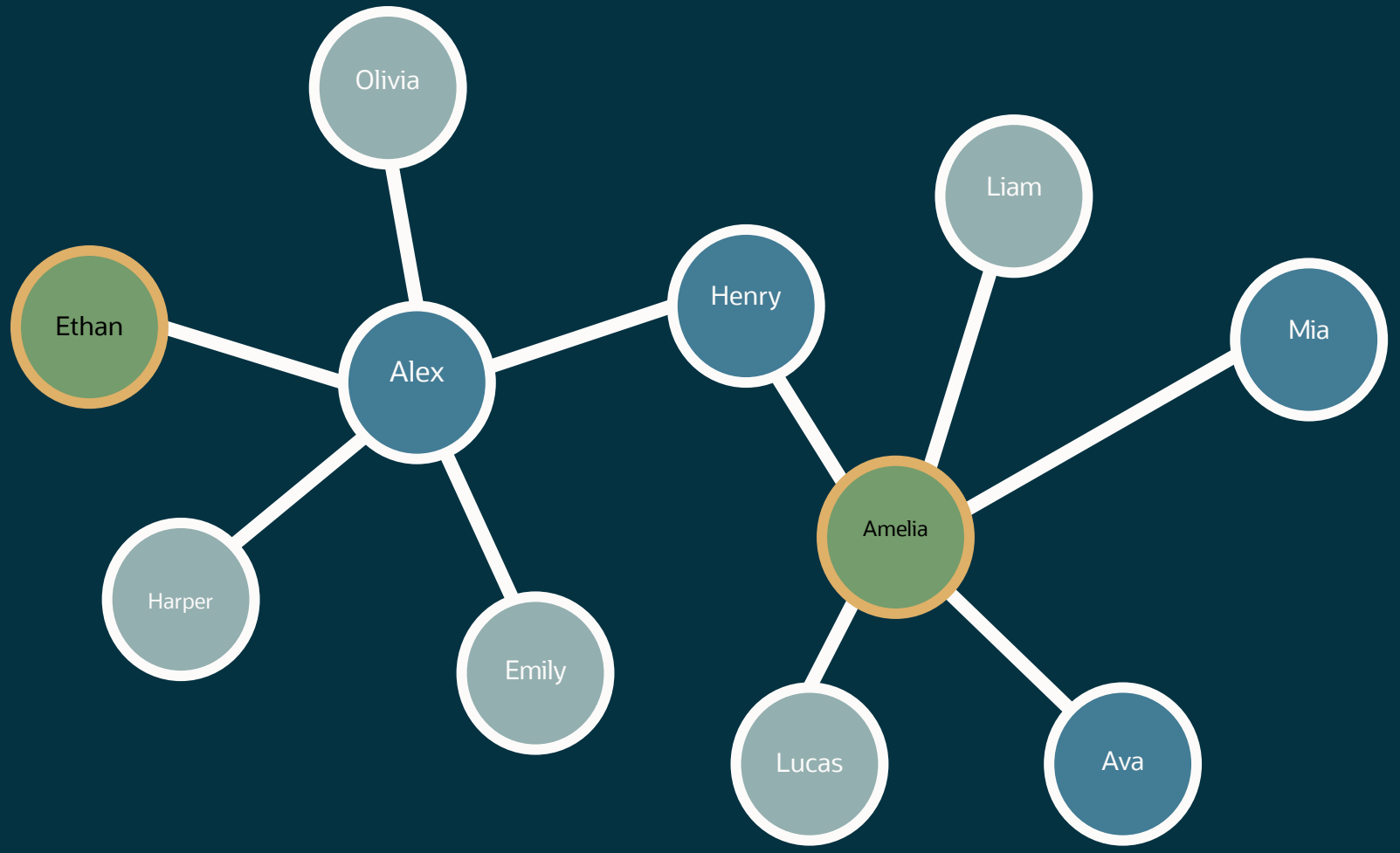
Graph data model captures how data is connected

What is a graph?

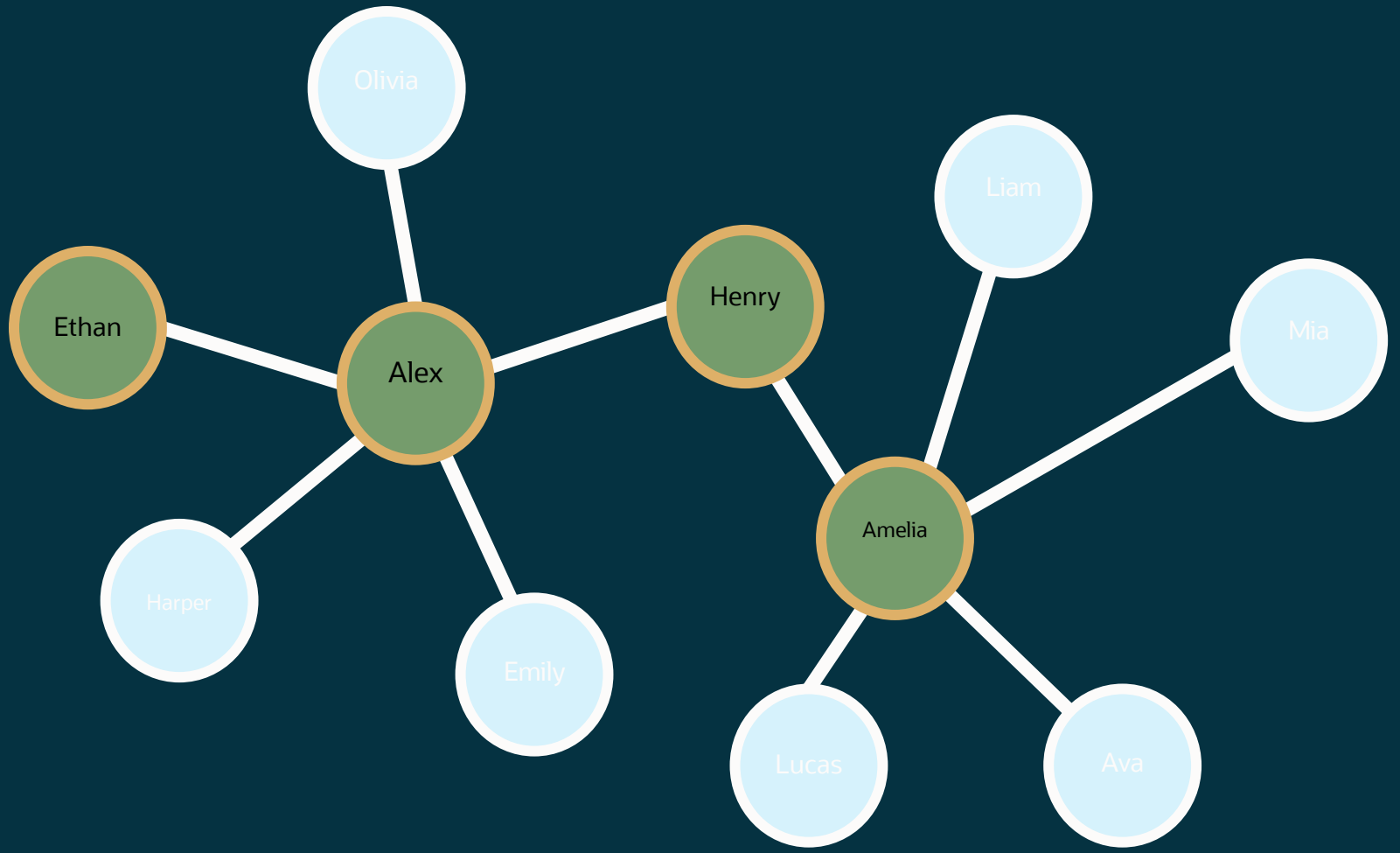
What is a graph?



Are you connected to someone you meet?



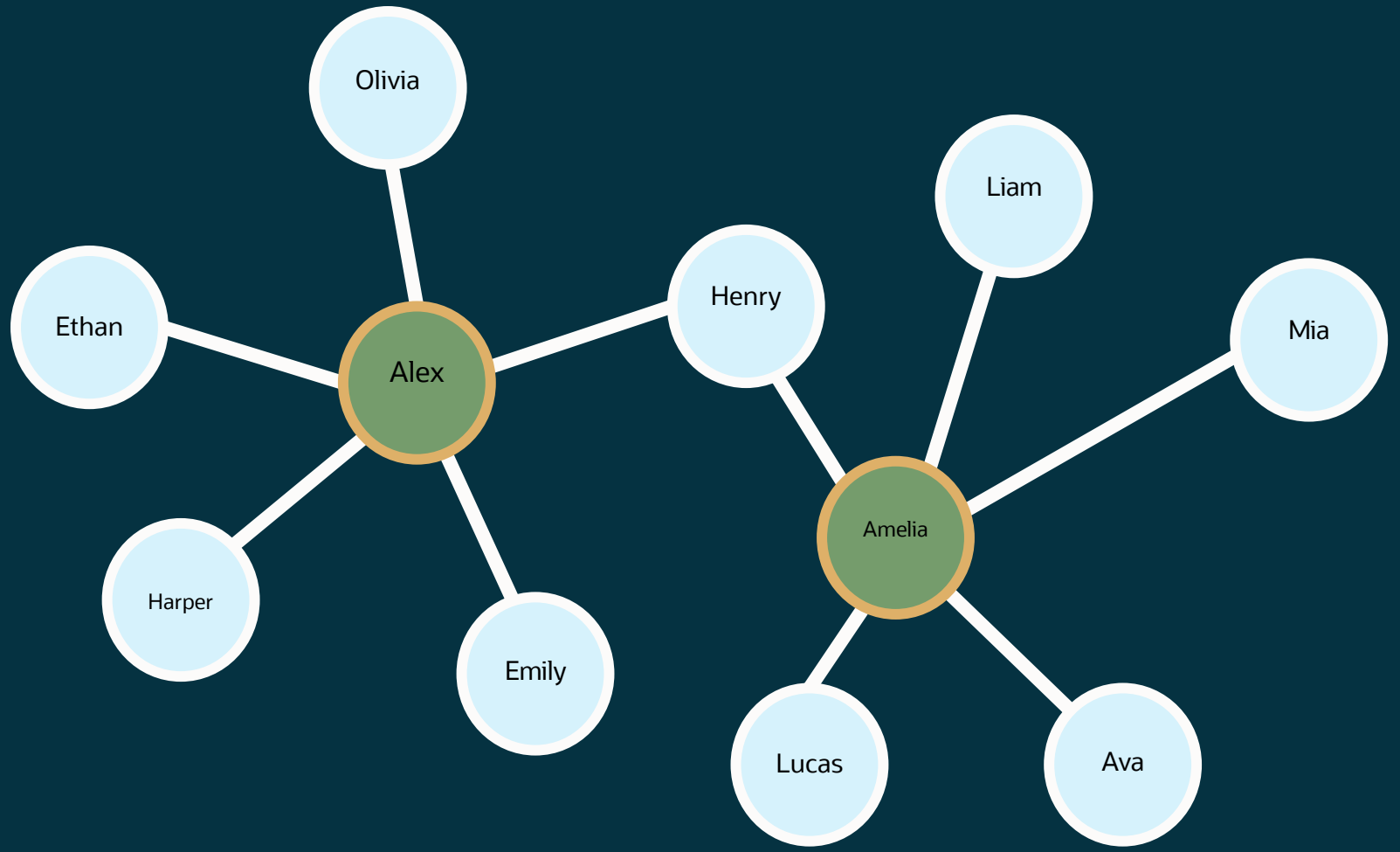
Are you connected to someone?



What is Ethan's connection to Amelia (number of hops and folks along the way)

Are there folks connected to a lot of people (like Alex and Amelia)

Who is highly connected?



What is Ethan's connection to Amelia (number of hops and folks along the way)

Are there folks connected to a lot of people (like Alex and Amelia)

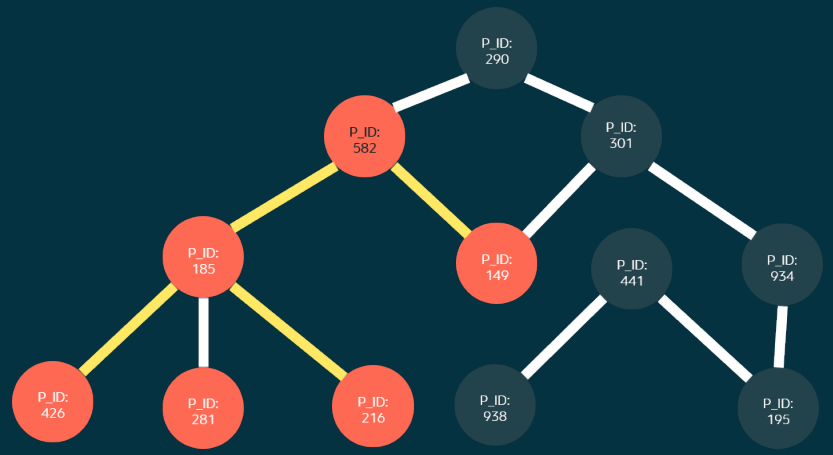
Social Networks are Natural Graphs

….. Many other types of data can be expressed as graphs as well, for new insights

Social Networks are Natural Graphs

….. Many other types of data can be expressed as graphs as well, for new insights

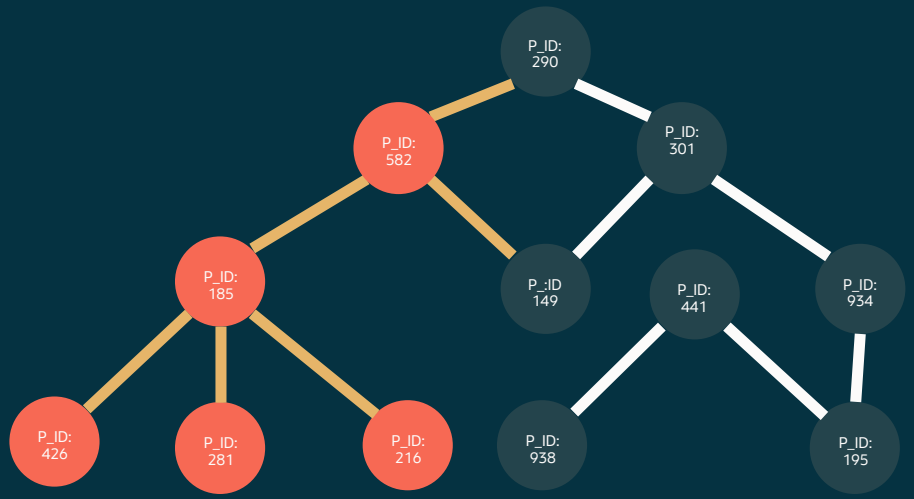
Analyze product dependencies in Manufacturing



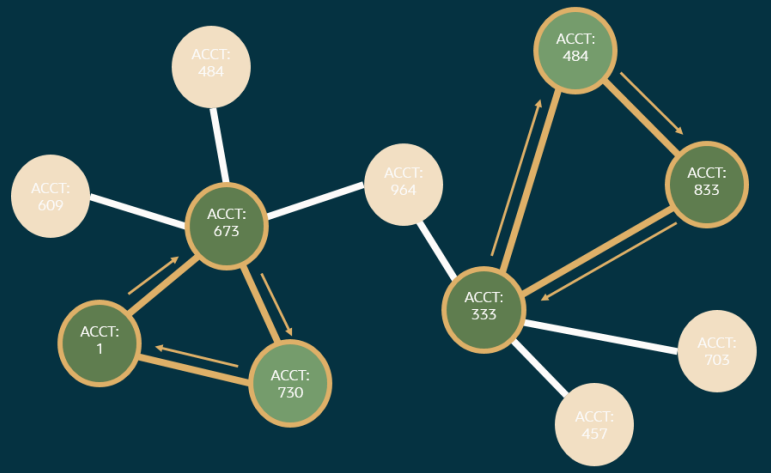
Social Networks are Natural Graphs

….. Many other types of data can be expressed as graphs as well, for new insights

Analyze product dependencies in Manufacturing



Find anomalous patterns in financial transactions



Ex: Are there money transfer patterns that are cycles?

In Oracle Database a Graph is like a View on Relational Tables

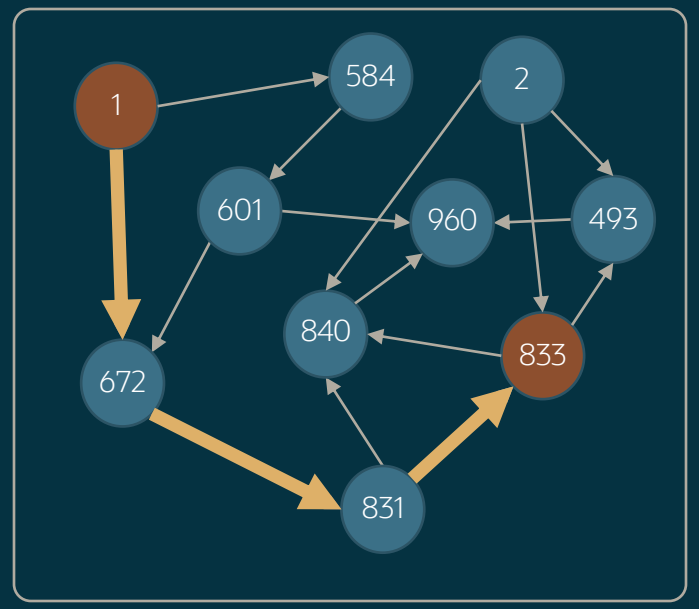
- No data copy – graph is a view on relational tables
- Insert/update/delete in underlying tables instantly available in a graph
- Fast, concurrent updates

BANK_ACCOUNTS

ID	NAME
1	
2	
3	
…	
672	
673	
674	
…	
831	
832	
833	
…	

BANK_TRANSFERS

SRC_ACCT_ID	DST_ACCT_ID	AMOUNT
1	672	1000
1	584	1000
1	259	100000
2	833	5001
2	840	7050
2	493	4363
…	…	…
672	831	5425
…	…	…
831	833	256
…	…	…



Graph Queries using SQL in Oracle Database 23ai

Find all accounts 934 has transferred cash to

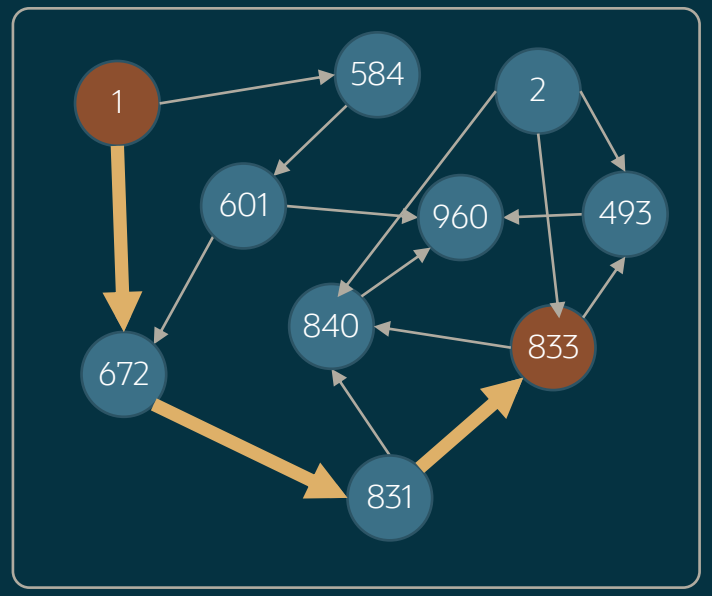
```
SELECT * from GRAPH_TABLE(bank_graph
MATCH (a) -[]->(b)
COLUMNS (a.id as acct_id, b.id as dst_id))
WHERE acct_id = 934;
```

Find all paths between 934 and 387,
via 1 to 4 cash transfers

```
SELECT * from GRAPH_TABLE(bank_graph
MATCH (a) -[]->{1,4}(b)
COLUMNS (a.id as acct_id, b.id as dst_id))
WHERE acct_id = 934 and dst_id = 387;
```

Is ID <src> connected to ID <dst> in 1 to 3 hops?

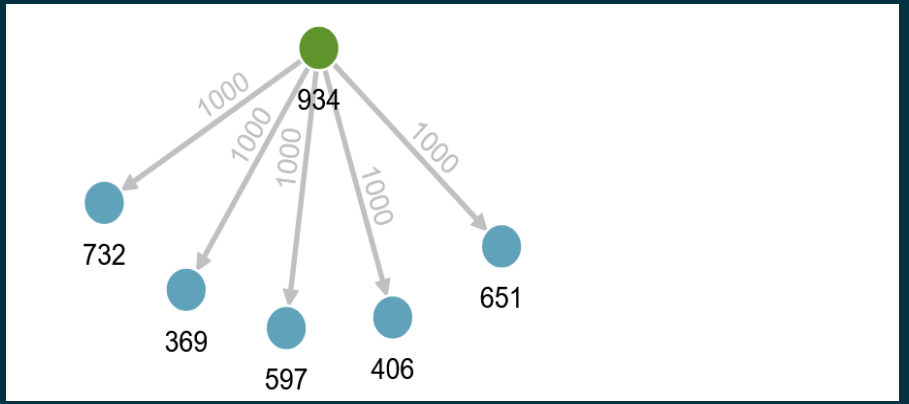
```
-- With new GRAPH_TABLE and MATCH syntax

SELECT account_id
   FROM GRAPH_TABLE(bank_graph
   MATCH (src)-[is bank_transfers]->{1,3}(dst)
   COLUMNS (src.id as account_id) );
```



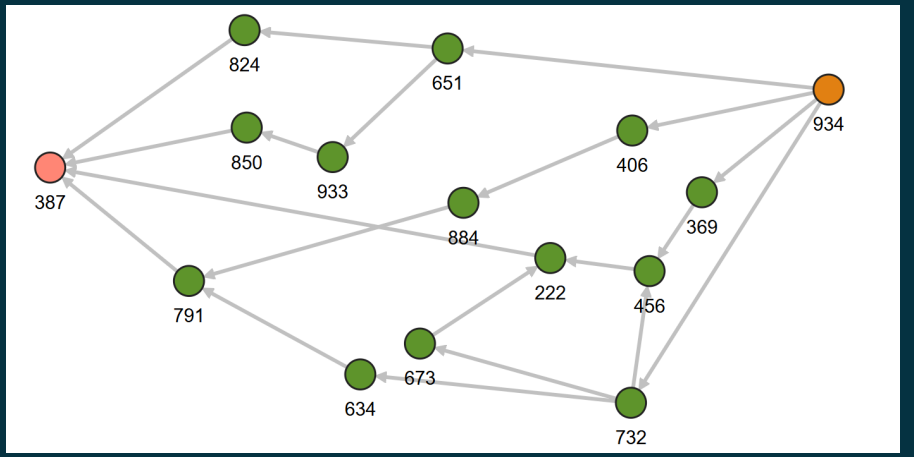

```
-- SQL without new syntax (12 joins and 3 UNION ALLs)

SELECT v1.id as account_id1 , v2.id as account_id2
FROM   bank_accounts  v1 ,
       bank_transfers  btx,
       bank_accounts  v2
WHERE  (v1.id = btx.src_acct_id  AND v2.id = btx.dst_acct_id)
AND     v1.id= <src> AND v2.id= <dst>
UNION ALL
SELECT v1.id as account_id1 , v2.id as account_id2
FROM   bank_accounts  v1 ,
       bank_transfers  btx,
       bank_accounts  bc2,
       bank_transfers  btx2 ,
       bank_accounts  v2
WHERE  (v1.id = btx.src_acct_id  AND  bc2.id = btx.dst_acct_id  AND
        bc2.id = btx2.src_acct_id  AND v2.id = btx2.dst_acct_id  )
AND     v1.id= <src> AND v2.id= <dst>
UNION ALL
SELECT v1.id as account_id1 ,v2.id as account_id2
FROM   bank_accounts  v1 ,
       bank_transfers  btx,
       bank_accounts  bc2,
       bank_transfers  btx2 ,
       bank_accounts  bac4,
       bank_transfers  btx5 ,
       bank_accounts  v2
WHERE  (v1.id = btx.src_acct_id  AND bc2.id = btx.dst_acct_id  AND
        bc2.id = btx2.src_acct_id  AND bac4.id = btx2.dst_acct_id  AND
        bac4.id = btx5.src_acct_id  AND  v2.id = btx5.dst_acct_id  )
AND     v1.id= <src> AND v2.id= <dst>
;
```

Graph Visualization

Find all accounts 934 has transferred cash to



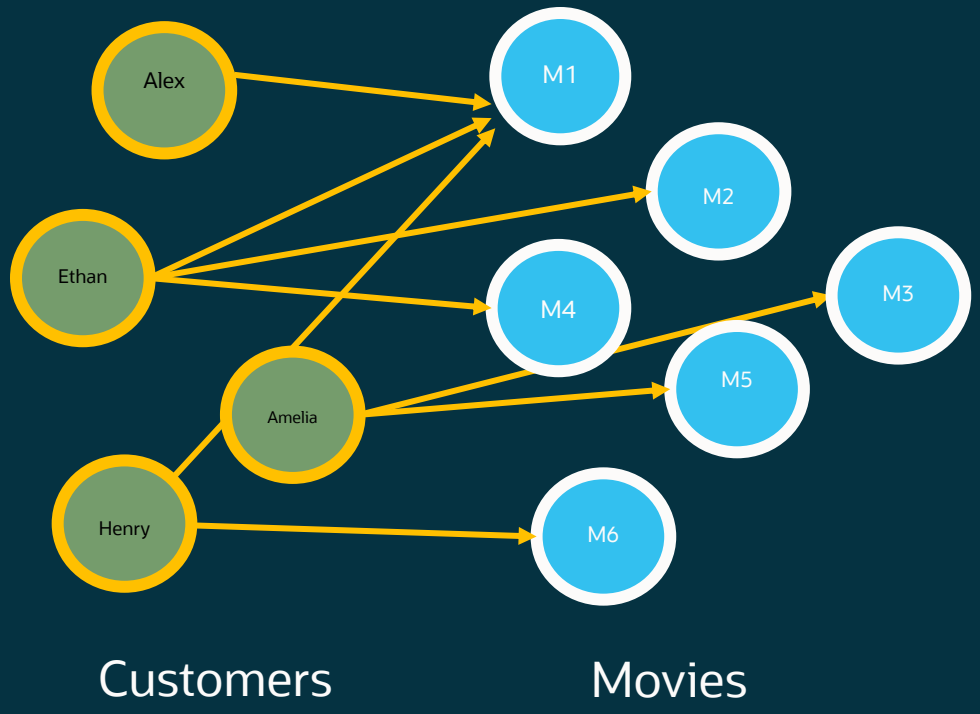
Find all paths between 934 and 387,
via 1 to 4 cash transfers



The MOVIEGRAPH Example

We will use this for some of our Graph RAG examples

Example Graph: Customers Watch Movies

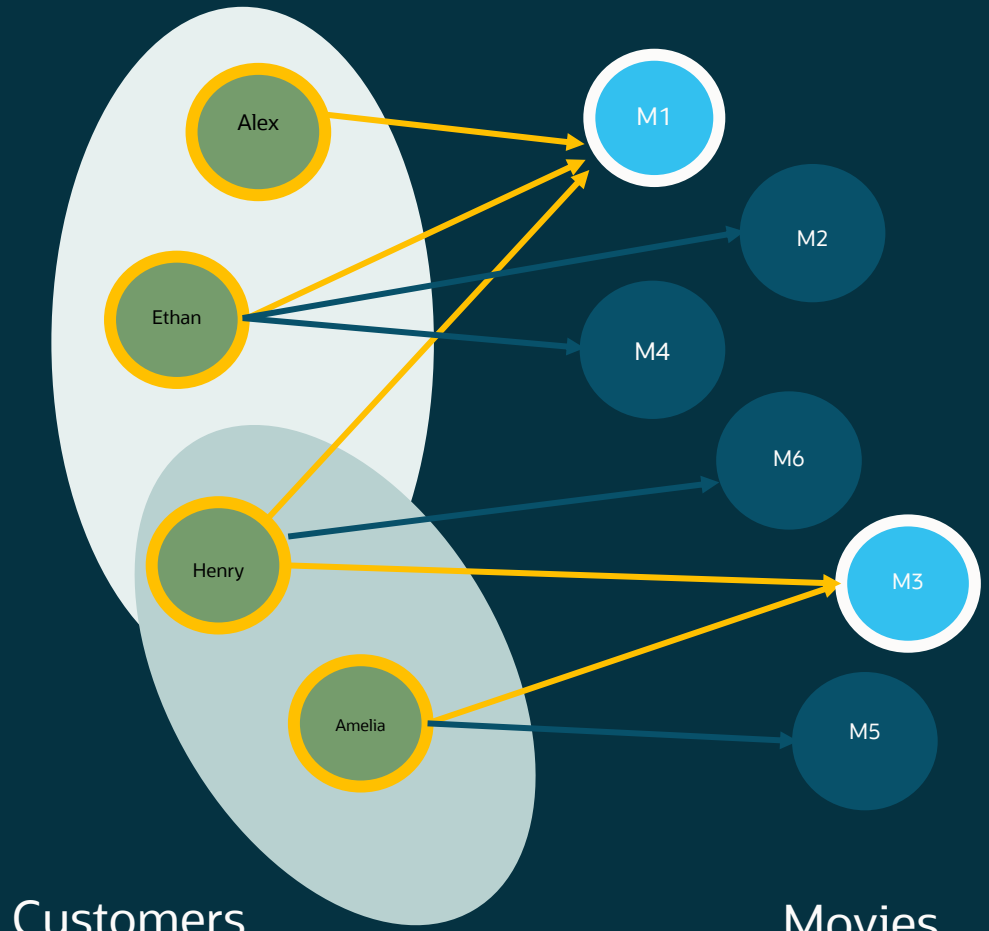


Relationship 1:
(CUSTOMER) -> (MOVIE)

Questions

- Which customers have similar movie watching patterns?

Customers Have Watch Parties

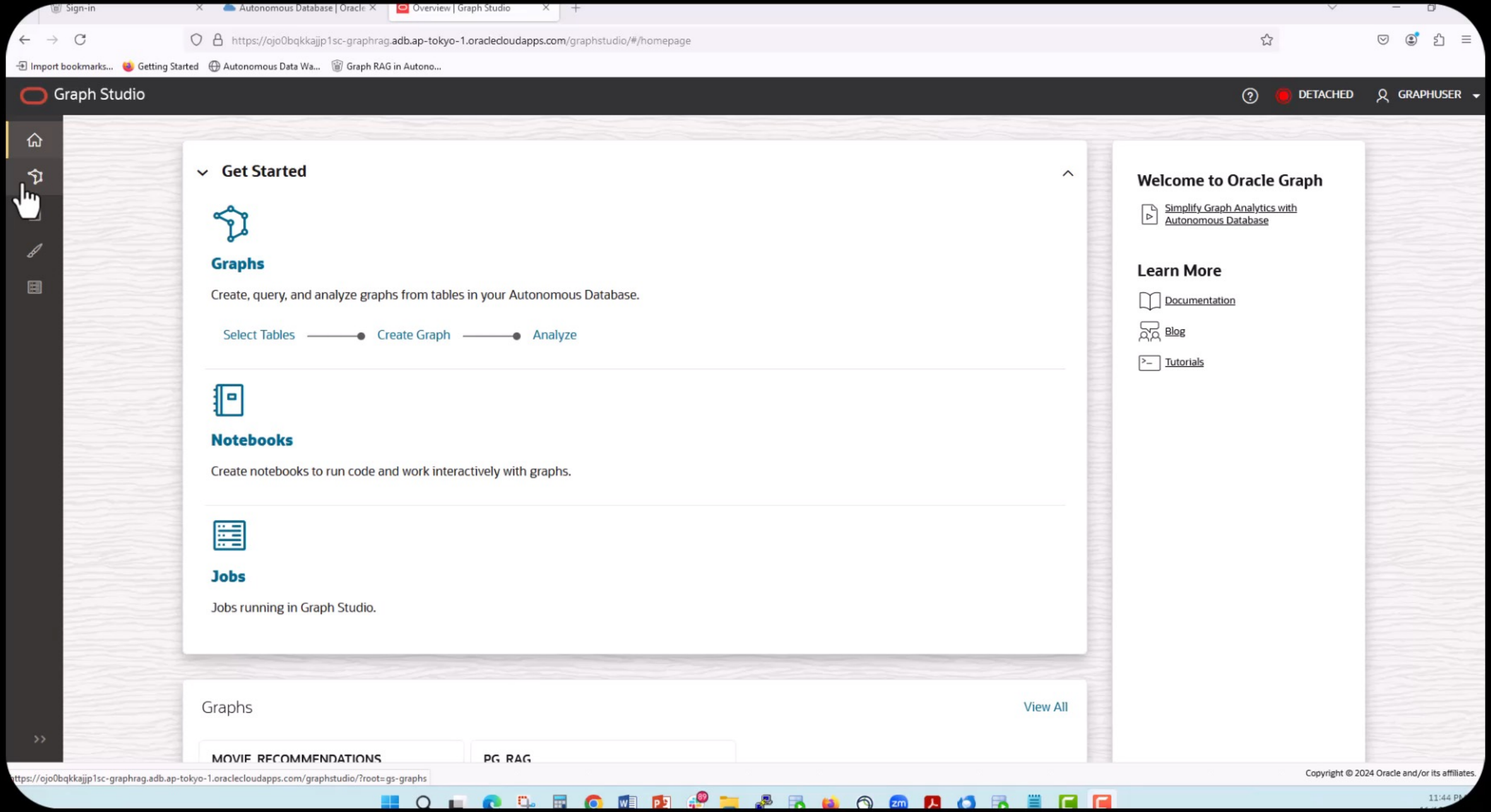


Relationship 2:
(CUSTOMER) -> (CUSTOMER) -> (MOVIE)

Questions

- Which customers are connected by watch parties?

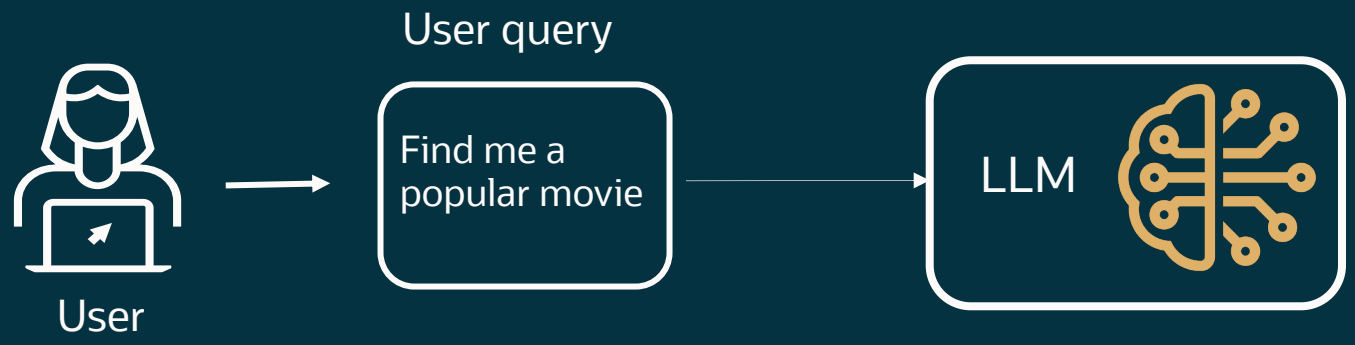
Creating a graph is very easy with Autonomous Database


Sign-in
Autonomous Database | Oracle
Overview | Graph Studio
https://ojo0bqkkajjp1sc-graphrag.adb.ap-tokyo-1.oraclecloudapps.com/graphstudio/#/homepage
Import bookmarks...
Getting Started
Autonomous Data Wa...
Graph RAG in Autono...
Graph Studio
DETACHED
GRAPHUSER
Get Started
Graphs
Create, query, and analyze graphs from tables in your Autonomous Database.
Select Tables
Create Graph
Analyze
Notebooks
Create notebooks to run code and work interactively with graphs.
Jobs
Jobs running in Graph Studio.
Welcome to Oracle Graph
Simplify Graph Analytics with Autonomous Database
Learn More
Documentation
Blog
Tutorials
Graphs
View All
MOVIE_RECOMMENDATIONS
PG_RAG
https://ojo0bqkkajjp1sc-graphrag.adb.ap-tokyo-1.oraclecloudapps.com/graphstudio/?root=gs-graphs
Copyright © 2024 Oracle and/or its affiliates.

Graph RAG

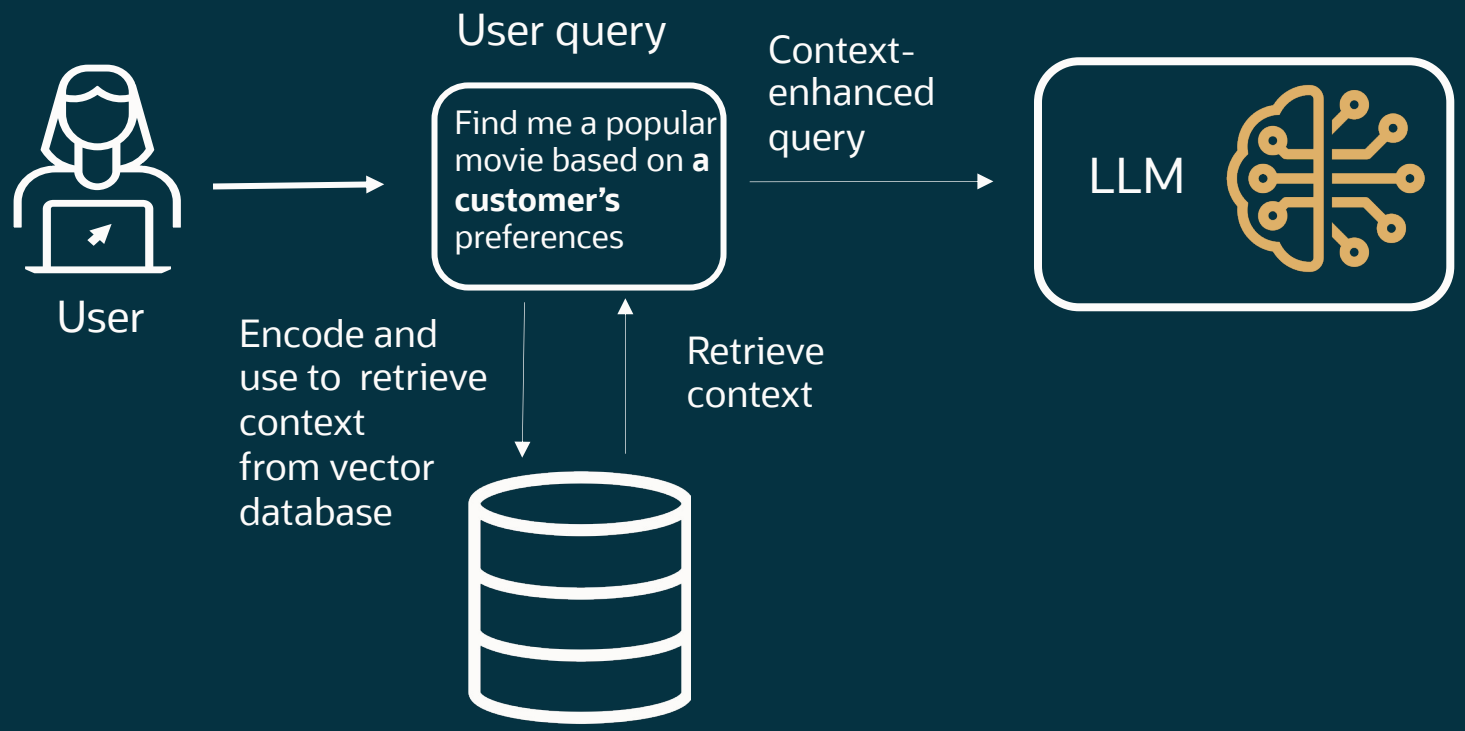
First, a RAG recap

Asking an LLM a Question



RAG: Retrieval Augmented Generation

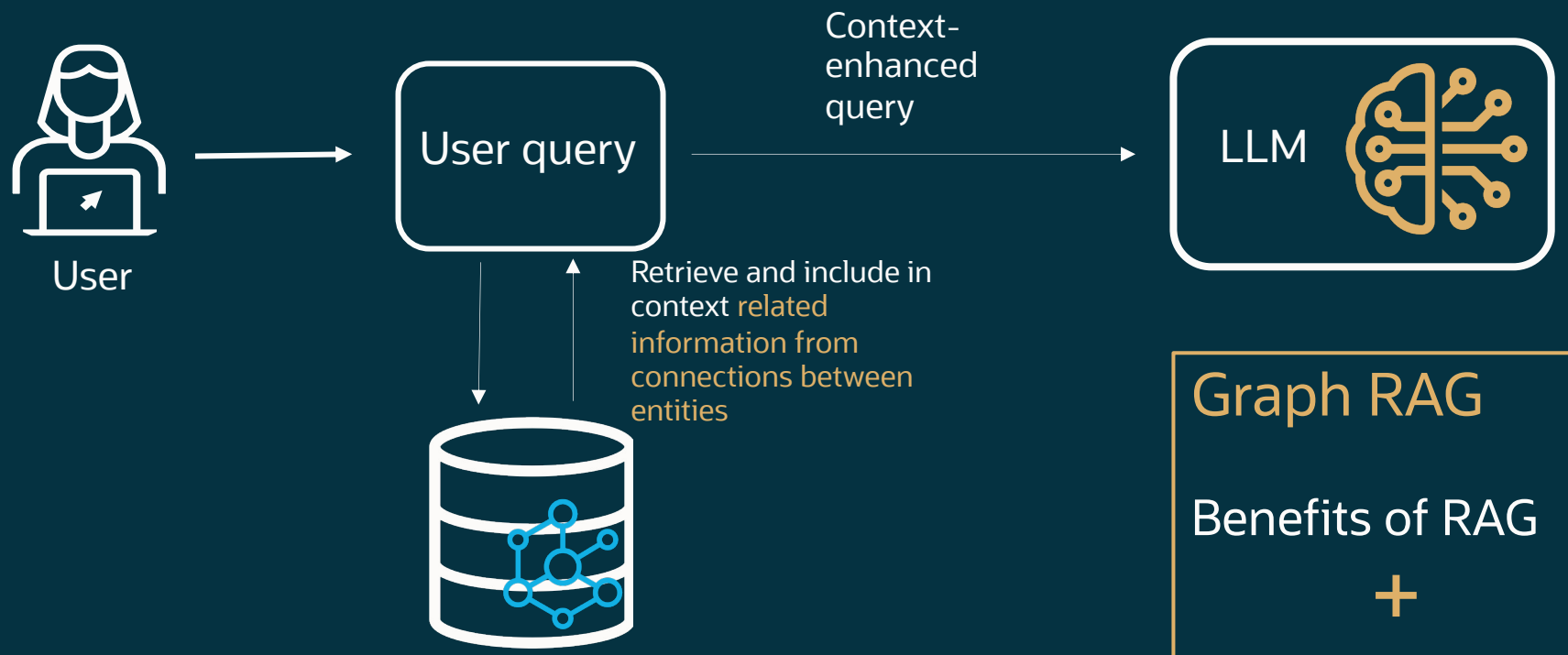
Enhance user query with data in the database



- Use latest and private data from a database to provide context to LLM
- Typically use vector search to find matching data in a database to provide as context to an LLM
  - Create embeddings for data and store as vectors in a vector database
  - Vectorize user's natural language query and match with stored vectors
  - Augment user query with top matches from the database

Graph RAG: Adding Graph Relationships to RAG

Enhance user query with data in the database



Graph RAG

Benefits of RAG

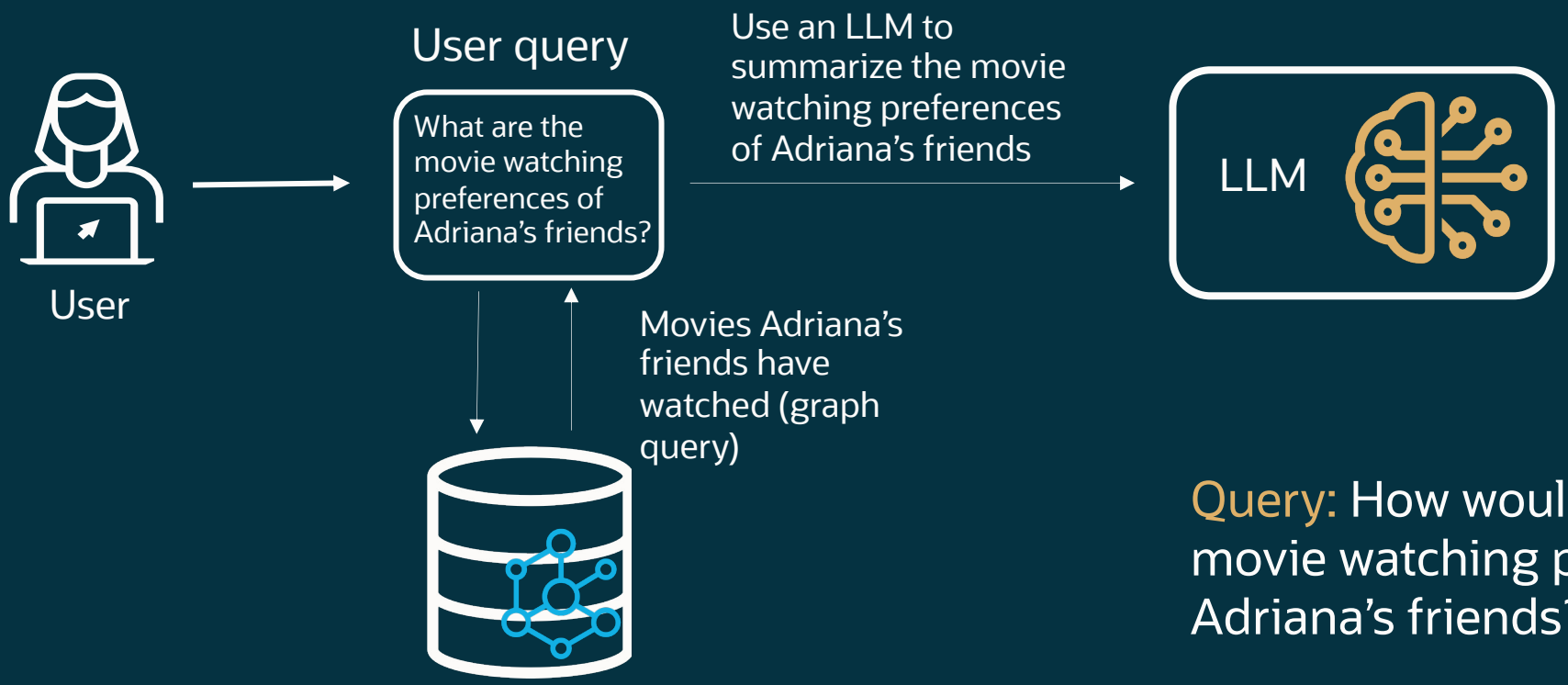
\+

Information on how data is connected

Example 1

Graph queries in LangChain

Summarizing Movie Watching Preferences with an LLM's help



Query: How would you describe the movie watching preferences of Adriana's friends?

GraphRag-Example1

localhost:8888/notebooks/GraphRag-Example1.ipynb

Graph | Graph Feature Requ... | Oracle Property Gra... | Graph Studio: Intera... | Oracle Support Portal | Graph Studio - 23ai | SQLDatabaseAction... | Graph Size Estimator | All Bookmarks

Jupyter GraphRag-Example1 Last Checkpoint: 17 minutes ago

Trusted

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipykernel)

[ ]:
```
from langchain_openai import ChatOpenAI

# Set up OpenAI LLM
print ("WARNING: The step will fail if the API key is not present or is incorrect.")
print ("Please update the OpenAI_API_key before calling the llm the next step.")

# set the LLM to get response
llm_gr = ChatOpenAI(
  model_name='gpt-3.5-turbo',
  temperature = 0.1,
  openai_api_key=".....",
  max_tokens=2000
)
print("The LLM model you will use is OpenAI ChatGPT 3.5")
```

[2]:
```
import oracledb
```

[3]:
```
import langchain_openai
```

[4]:
```
connection_gr =oracledb.connect(
     config_dir=r"C:\Users\Public\Downloads\Wallet_GraphRAG",
     user="graphuser",
     password="....",
     dsn="graphrag_medium",
     wallet_location=r"C:\Users\Public\Downloads\Wallet_GraphRAG",
     wallet_password="....")
```

[5]:
```
cursorgr = connection_gr.cursor()
```

[12]:
```
from langchain_core.prompts import PromptTemplate

# Create a prompt template
template = "What is the genre of {movie} based on this summary: {summary}?"
prompt = PromptTemplate.from_template(template)

# Create a chain with the prompt and LLM
chain = prompt | llm_gr
```

[ ]:
```
# Define your SQL query
sql = """
    SELECT DISTINCT MOVIE_TITLE, MOVIE_SUMMARY
```

Using LangChain

Blog post on Medium.com

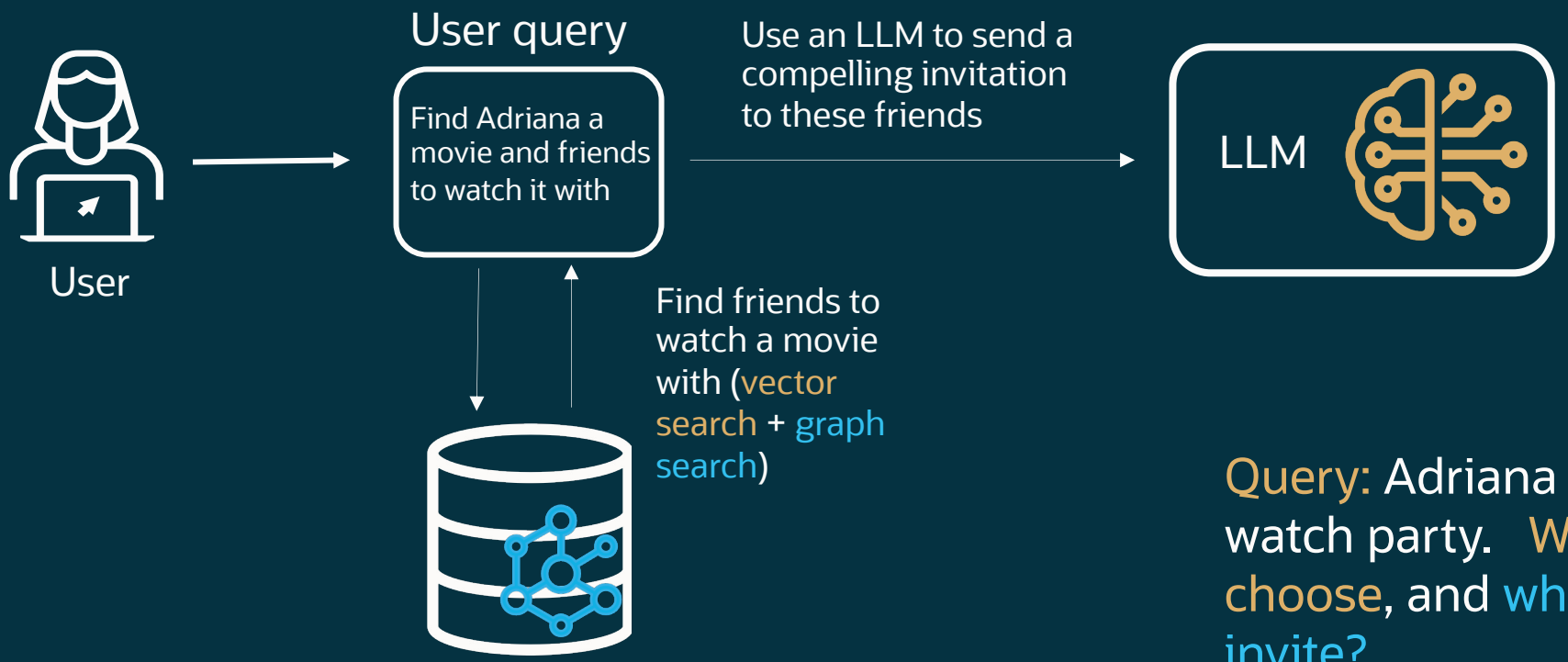
Use graph query in Autonomous Database with LangChain

```
# Define your SQL query
sql = """
    SELECT DISTINCT MOVIE_TITLE, MOVIE_SUMMARY
    FROM GRAPH_TABLE( MOVIE_RECOMMENDATIONS
        MATCH (c1 is CUSTOMER) -[e is WATCHED_WITH]-> (c2 is CUSTOMER)-[w is WATCHED]-> (m is MOVIE)
        WHERE c1.FIRST_NAME = 'Adriana' and c1.LAST_NAME = 'Osborne'
        COLUMNS (m.title as MOVIE_TITLE, m.summary as MOVIE_SUMMARY)
    ) FETCH FIRST 3 ROWS ONLY
"""

# Execute the SQL query

cursor2.execute(sql)

#for row in cursor2.execute(sql):
#   print(row)

# Fetch all rows from the executed query
rows = cursor2.fetchall()
# print(rows)

# Print the results
for row in rows:
#    Run the chain and print the output
   result = chain.invoke({'movie': row[0], 'summary': row[1]})
   print(result.content)
```

Example 2

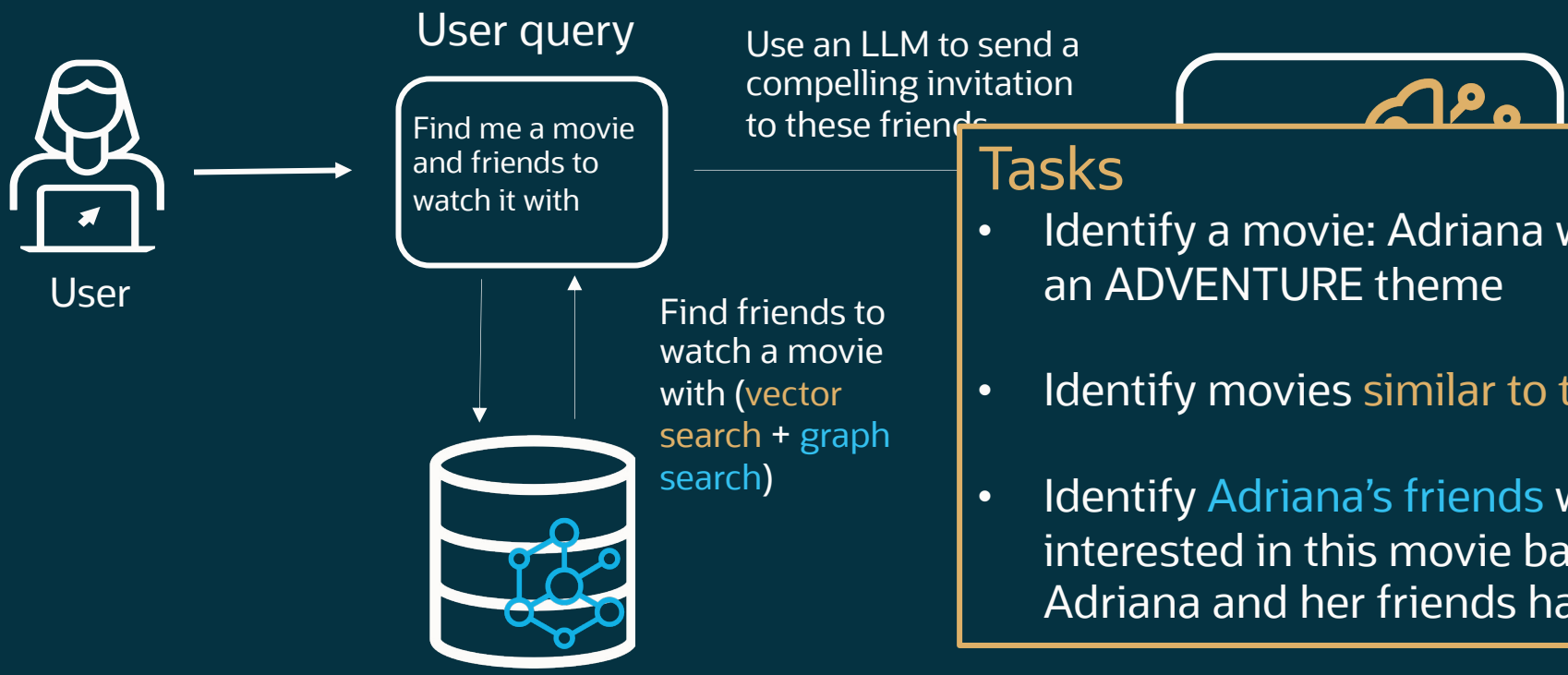
Combining Vector and Graph Search

Combining Vector and Graph Search



Query: Adriana wants to create a watch party. What movie should she choose, and which friends should she invite?

Combining Vector and Graph Search



Tasks

- Identify a movie: Adriana wants a 2024 movie with an ADVENTURE theme
- Identify movies similar to this movies
- Identify Adriana's friends who are likely to be interested in this movie based on similar movies Adriana and her friends have watched together

Combining Vector and Graph Search

User

User query

Find me a movie and friends to watch it with

Use an LLM to send a compelling invitation

Find fr
watch
with (v
search
search

Tasks

- Identify a movie: Adriana wants a 2024 movie with an ADVENTURE theme

```
SELECT m.TITLE, m.MOVIE_ID, m.YEAR
FROM MOVIE m
WHERE m.YEAR = 2024
ORDER BY vector_distance(m.summary_vec,
vector_embedding(doc_model_bert using 'ADVENTURE' as data), COSINE)
```

Combining Vector and Graph Search

User

User query

Find me a movie and friends to watch it with

Use an LLM to send a compelling invitation to these friends

Find friends to watch a movie with (vector search + graph search)

Tasks

- Identify a movie: Adriana wants a 2024 movie with an ADVENTURE theme
- Identify movies similar to this movie

```
SELECT mo.TITLE, mo.YEAR, vector_distance(m.summary_vec,
mo.summary_vec, COSINE) as vec_dist
FROM MOVIE m, MOVIE mo
WHERE m.TITLE = 'The Fall Guy'
ORDER BY vec_dist;
```

Combining Vector and Graph Search

User

User query

Find me a movie and friends to watch it with

Use an LLM compelling to these fri

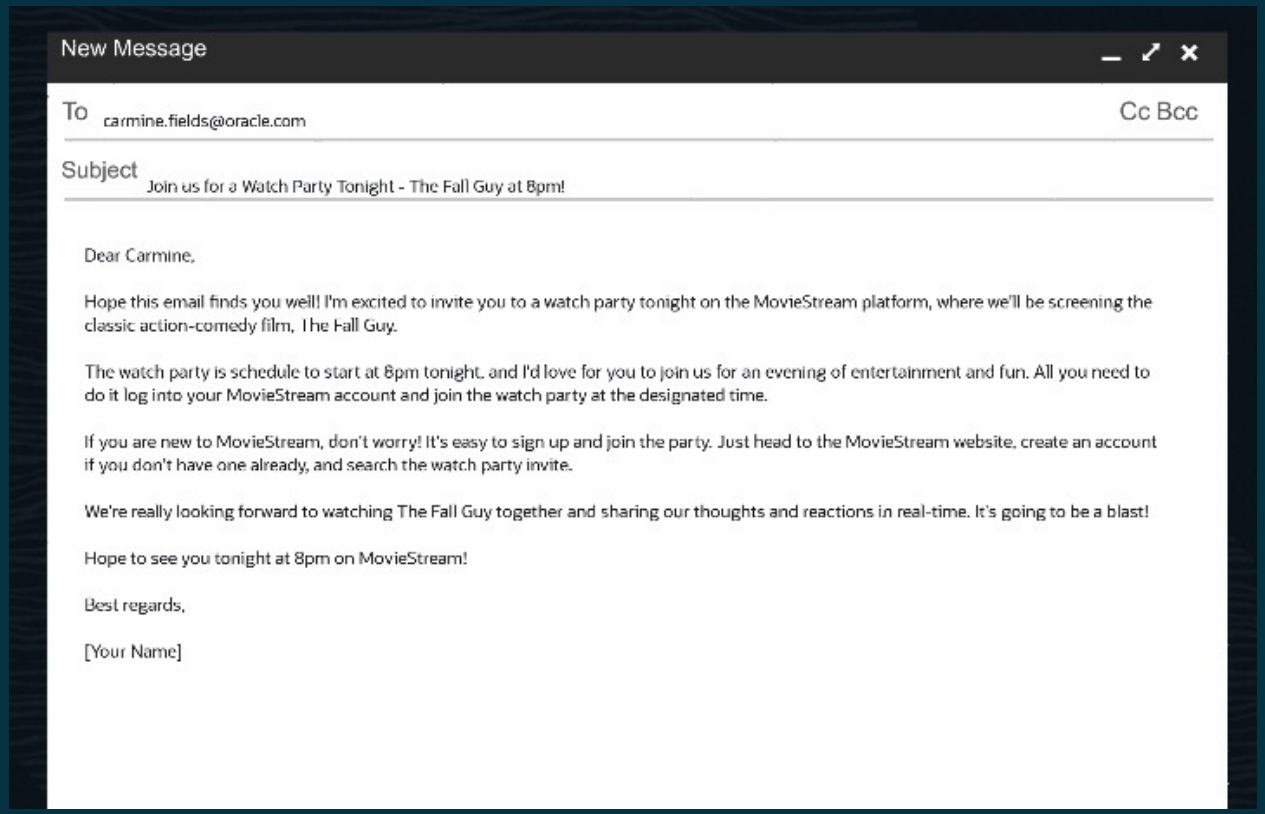
Find friends to watch a movie with (vector search + graph

Tasks

- Identify a movie: Adriana wants a 2024 movie with an ADVENTURE theme
- Identify similar movies to this movie
- Identify Adriana's friends who are likely to be interested in this movie based on similar movies Adriana and her friends have watched together

```
SELECT C1NAME, C2NAME, MOVIE_TITLE,
vector_distance(svec, (SELECT summary_vec from MOVIE WHERE title = 'The Fall Guy'), COSINE) as vec_dist
FROM (
    SELECT C1NAME, C2NAME, MOVIE_TITLE, svec
    FROM GRAPH_TABLE(MOVIE_RECOMMENDATIONS
        MATCH (c1 is CUSTOMER) -[e is WATCHED_WITH]-> (c2 is CUSTOMER)-[w is WATCHED]-> (m is MOVIE)
        WHERE e.MOVIE_ID = m.MOVIE_ID AND c1.FIRST_NAME ='Adriana' AND c1.LAST_NAME = 'Osborne'
        COLUMNS (…)
    )
)
ORDER BY vec_dist;
```

Send the invitee list to an LLM to generate invitations



MOVIE_RECOMMENDATIONS | SQL | Oracle Database Actions

https://ojo0bqkkajjp1sc-graphrag.adb.ap-tokyo-1.oraclecloudapps.com/graphstudio/#/notebook/dsVBqOgO

Import bookmarks... | Getting Started | Autonomous Data Wa... | Graph RAG in Autono... | Graph RAG Graph Stu...

Graph Studio

ATTACHED | GRAPHUSER

Notebooks > MOVIE_RECOMMENDATIONS

Versioning | Default | Zeppelin | Default Template

find movies with genre type 'Adventure' for the watch party.

```
%sql
SELECT m.TITLE, m.MOVIE_ID, m.YEAR
FROM MOVIE m
WHERE m.YEAR = 2024
ORDER BY vector_distance(m.summary_vec,
vector_embedding(doc_model_bert using 'ADVENTURE' as data), COSINE)
```

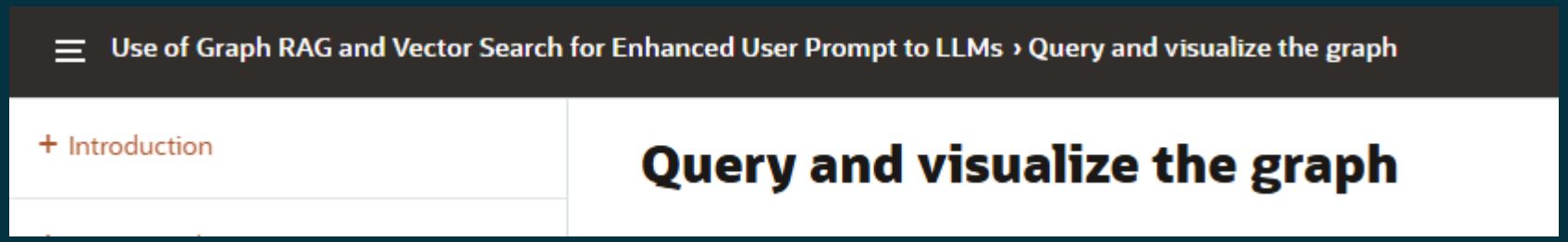
Our search result gives 'The Fall Guy' the highest score for 'Adventure' movies in 2024 that we have available for our customers to watch.

Adriana wants to know which movies it's similar to to help her decide if she'd like to watch it. Let's run another vector search to find out.

```
%sql
SELECT mo.TITLE, mo.YEAR, vector_distance(m.summary_vec, mo.summary_vec, COSINE) as vec_dist
FROM MOVIE m, MOVIE mo
WHERE m.TITLE = 'The Fall Guy'
ORDER BY vec_dist;
```

Task 2: Create and query a Property Graph using SQL

LiveLab: Combining Vector and Graph Search



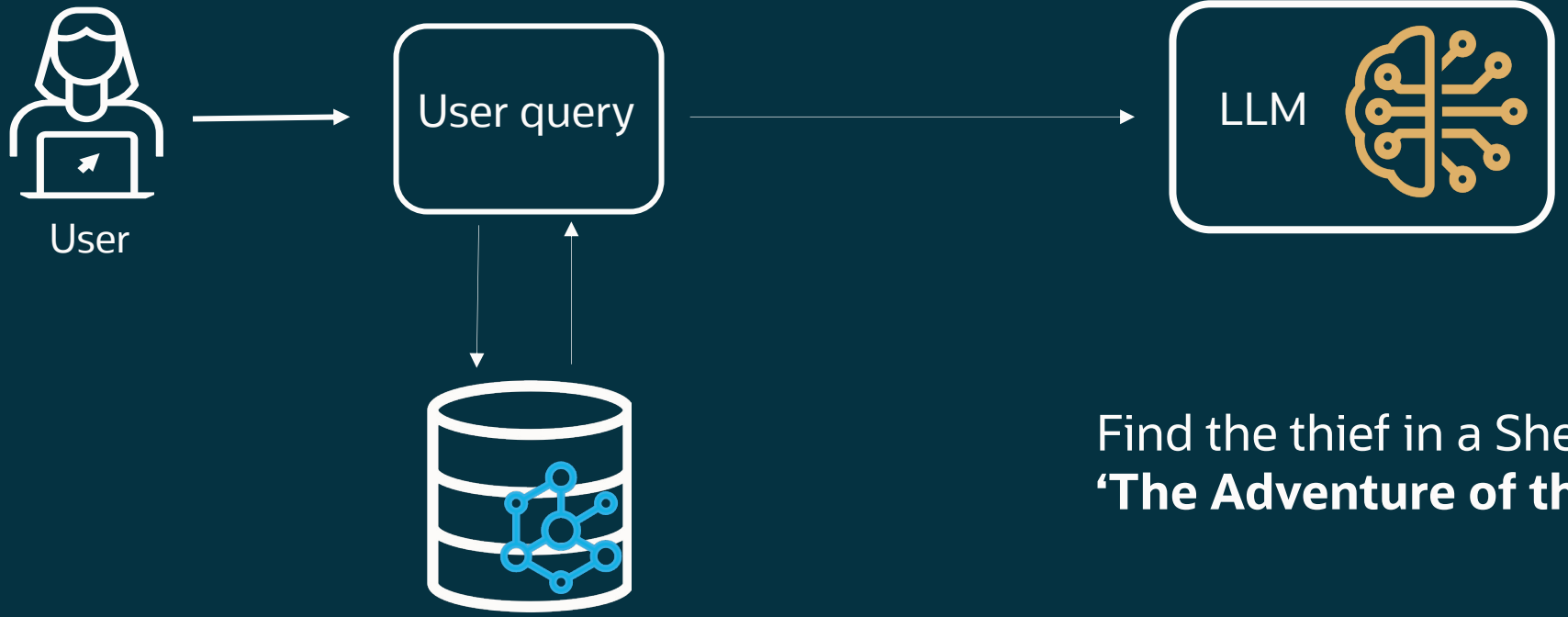
LiveLab: Visit livelabs.oracle.com and search for ‘graph rag’

Combine Vector and Graph search and call out to OCI gen AI service from Graph Studio in Autonomous Database

Example 3

Possibilities with Graph RAG

Example 3: Using Graph RAG to Improve gen AI Answers



Find the thief in a Sherlock Holmes Story
‘The Adventure of the Blue Carbuncle’

The Adventure of the Blue Carbuncle

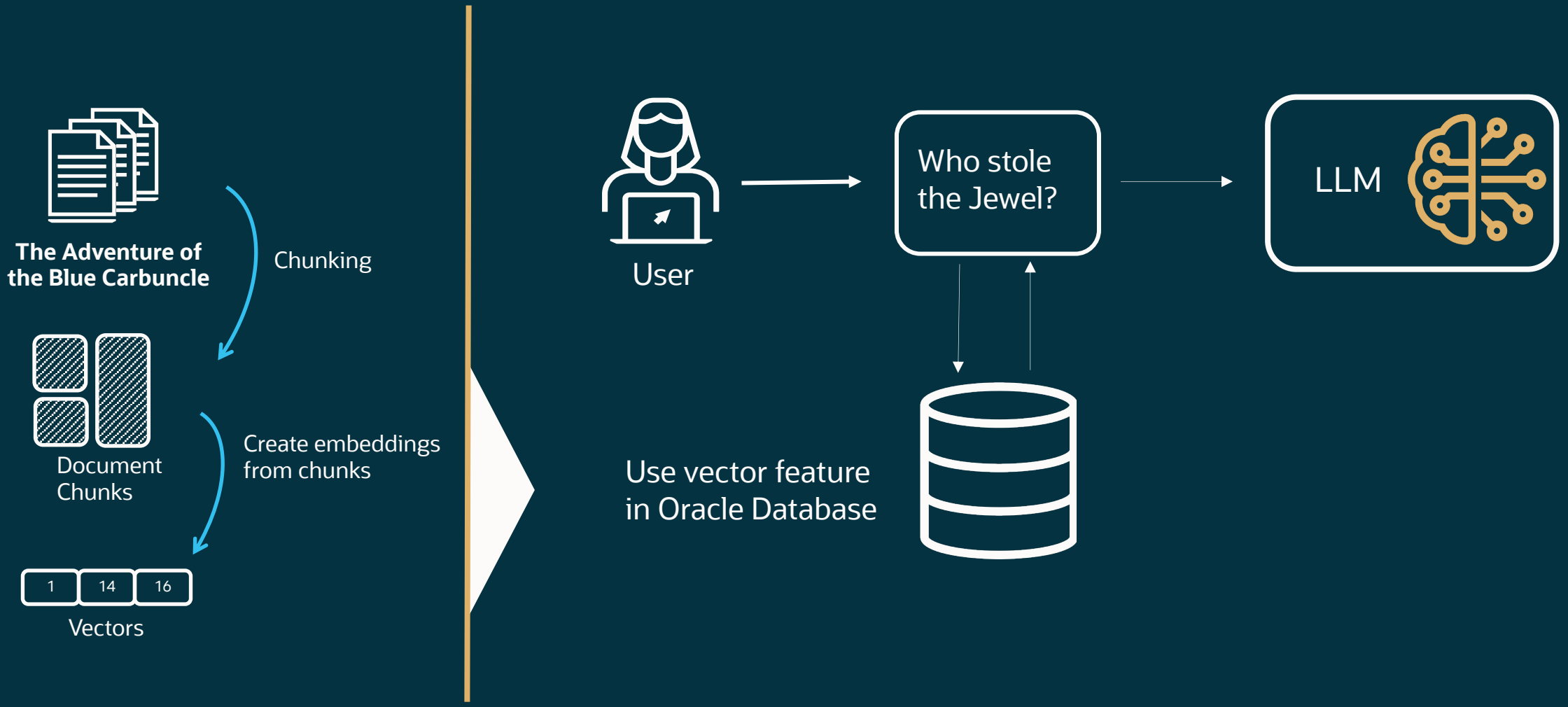
I had called upon my friend Sherlock Holmes upon the second morning after Christmas, with the intention of wishing him the compliments of the season. He was lounging upon the sofa in a purple dressing-gown, a pipe-rack within his reach upon the right, and a pile of crumpled morning papers, evidently newly studied, near at hand. Beside the couch was a wooden chair, and on the angle of the back hung a very seedy and disreputable hard-felt hat, much the worse for wear, and cracked in several places. A lens and a forceps lying upon the seat of the chair suggested that the hat had been suspended in this manner for the purpose of examination.

"You are engaged," said I; "perhaps I interrupt you."

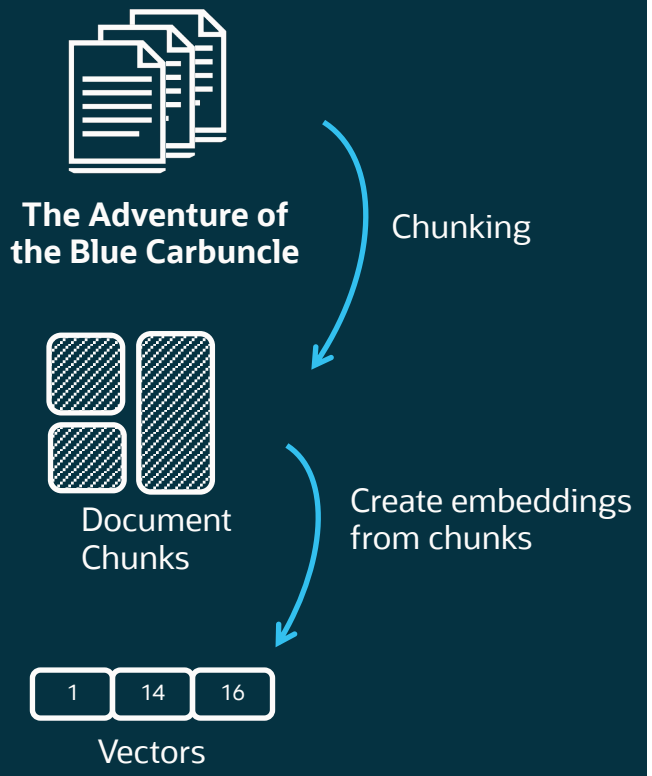
"Not at all. I am glad to have a friend with whom I can discuss my results. The matter is a perfectly trivial one" -- he jerked his thumb in the direction of the old hat --"but there are points in connection with it which are not entirely devoid of interest and even of instruction."

I seated myself in his armchair and warmed my hands before his crackling fire, for a sharp frost had set in, and the windows were thick with the ice crystals. "I suppose," I remarked, "that, homely as it looks, this thing has some deadly story linked on to it -- that it is the clew which will guide you in the solution of some mystery and the punishment of some crime."

Create Text Chunks and Vector Embeddings for RAG



Query Text Chunks with Vector Search



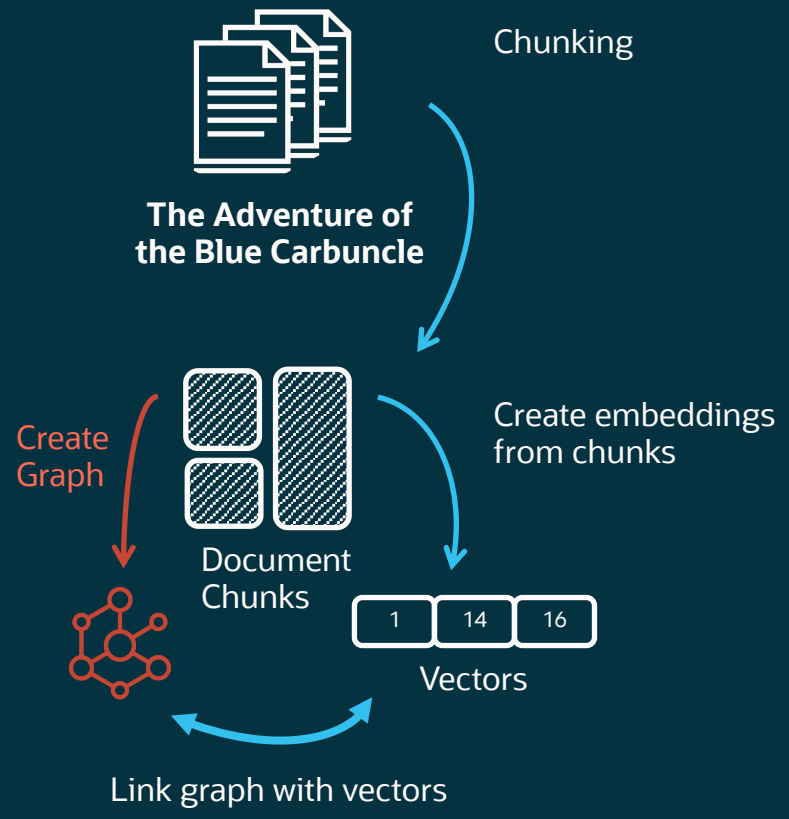
lady's jewel-case. The evidence against him was so strong that the case has been referred to the Assizes.

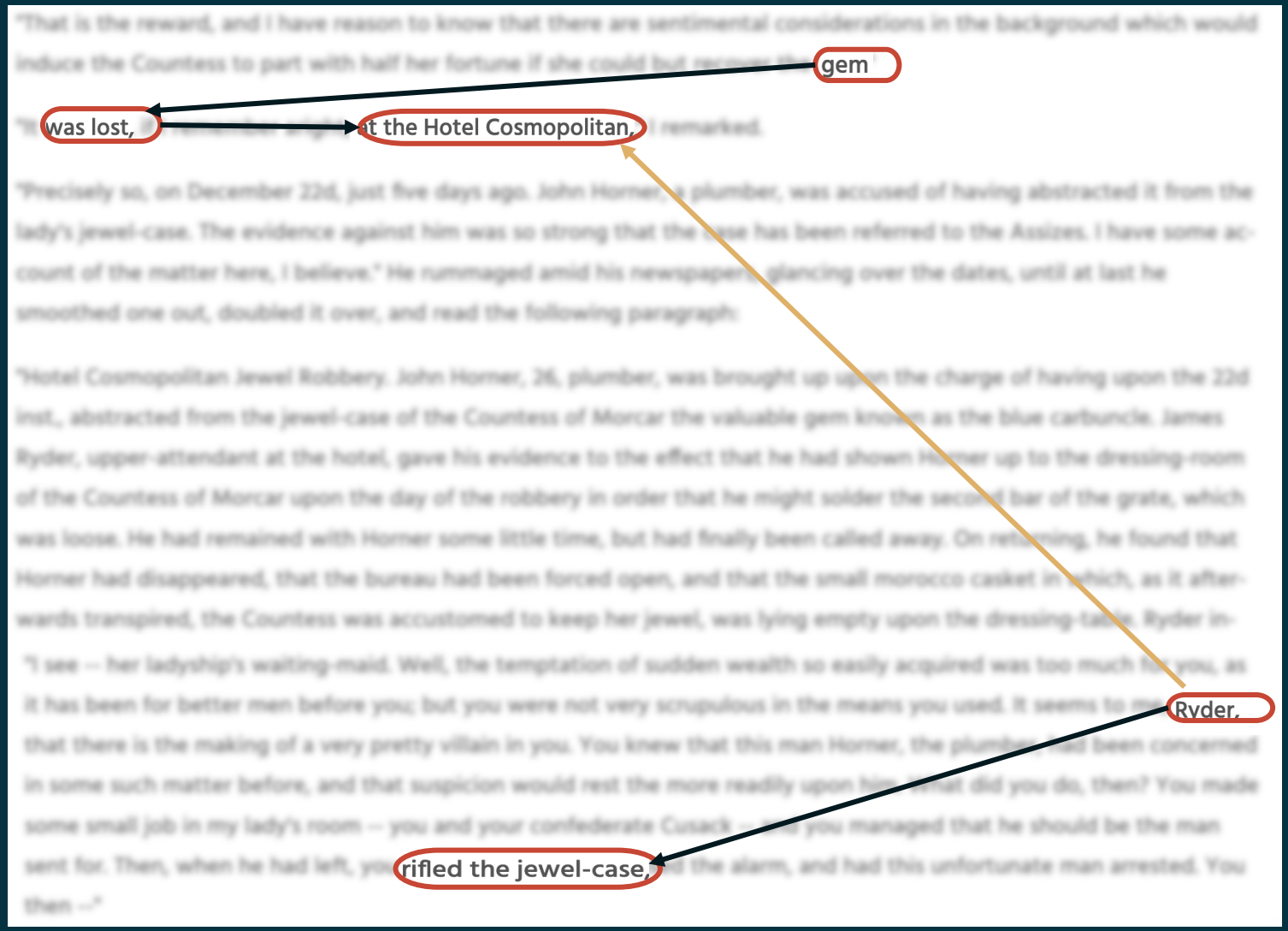
John Horner, 26, plumber, was brought up upon the charge of having upon the 22d inst., abstracted from the jewel-case of the Countess of Morcar the valuable gem known as the blue carbuncle.

Question: **Who stole the Jewel?**

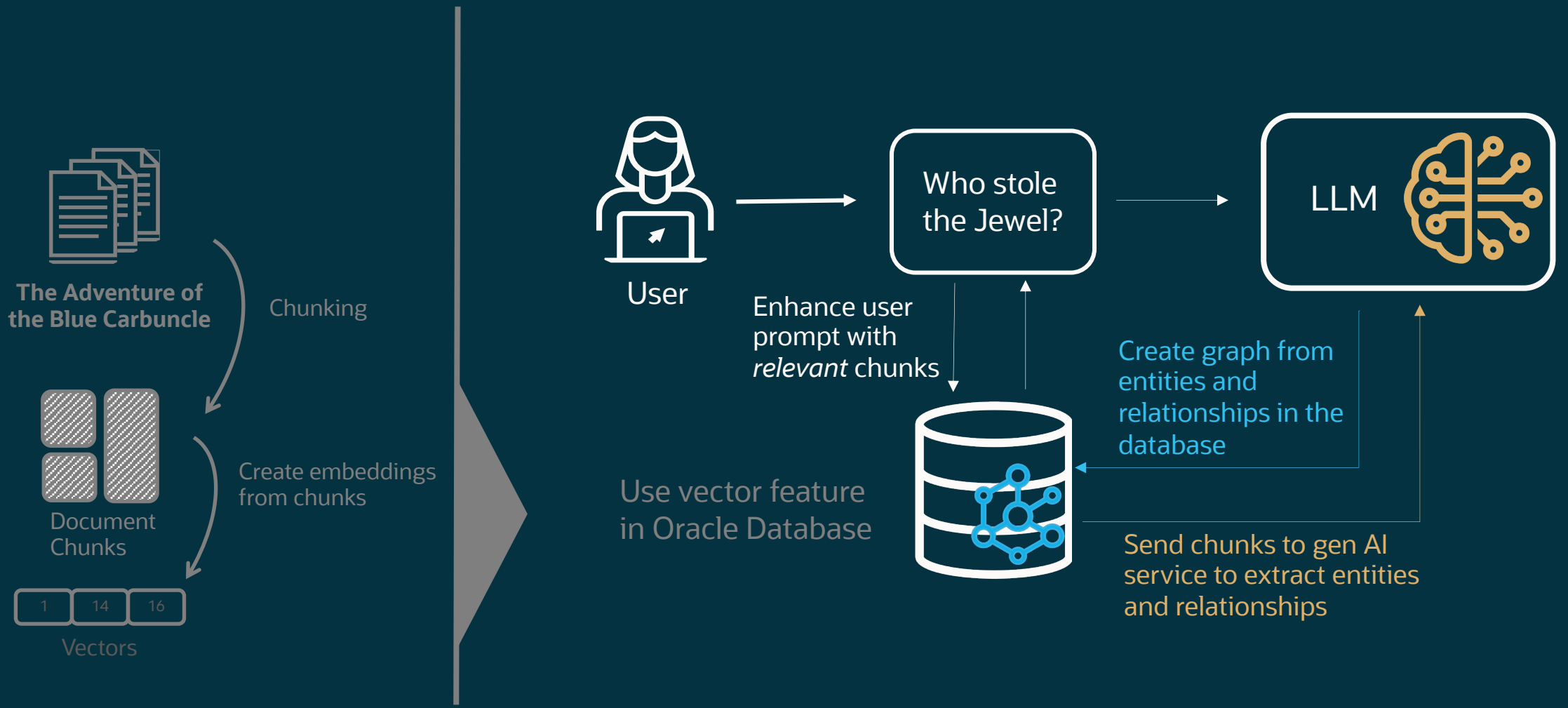
Answer: **The jewel was stolen by** John Horner, a **plumber, who was accused of ...**

Extract Graph and Link with Text Chunks

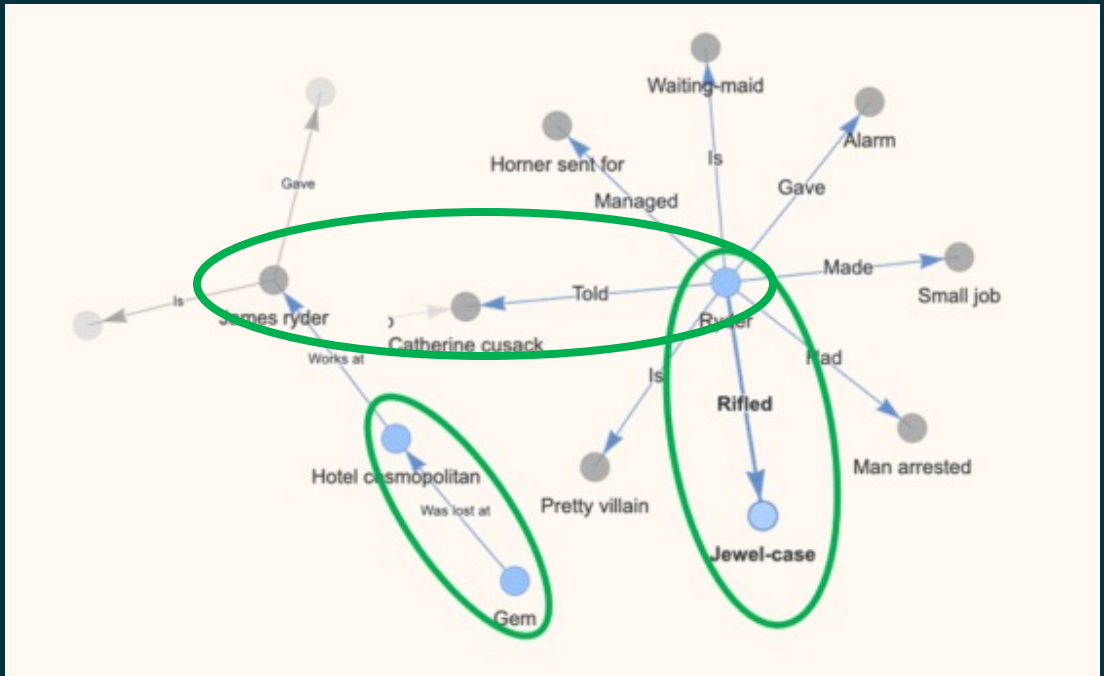




Extract Entities and Relationships for each Text Chunk



Graphs Connect Entities Across Text Chunks



Graph RAG

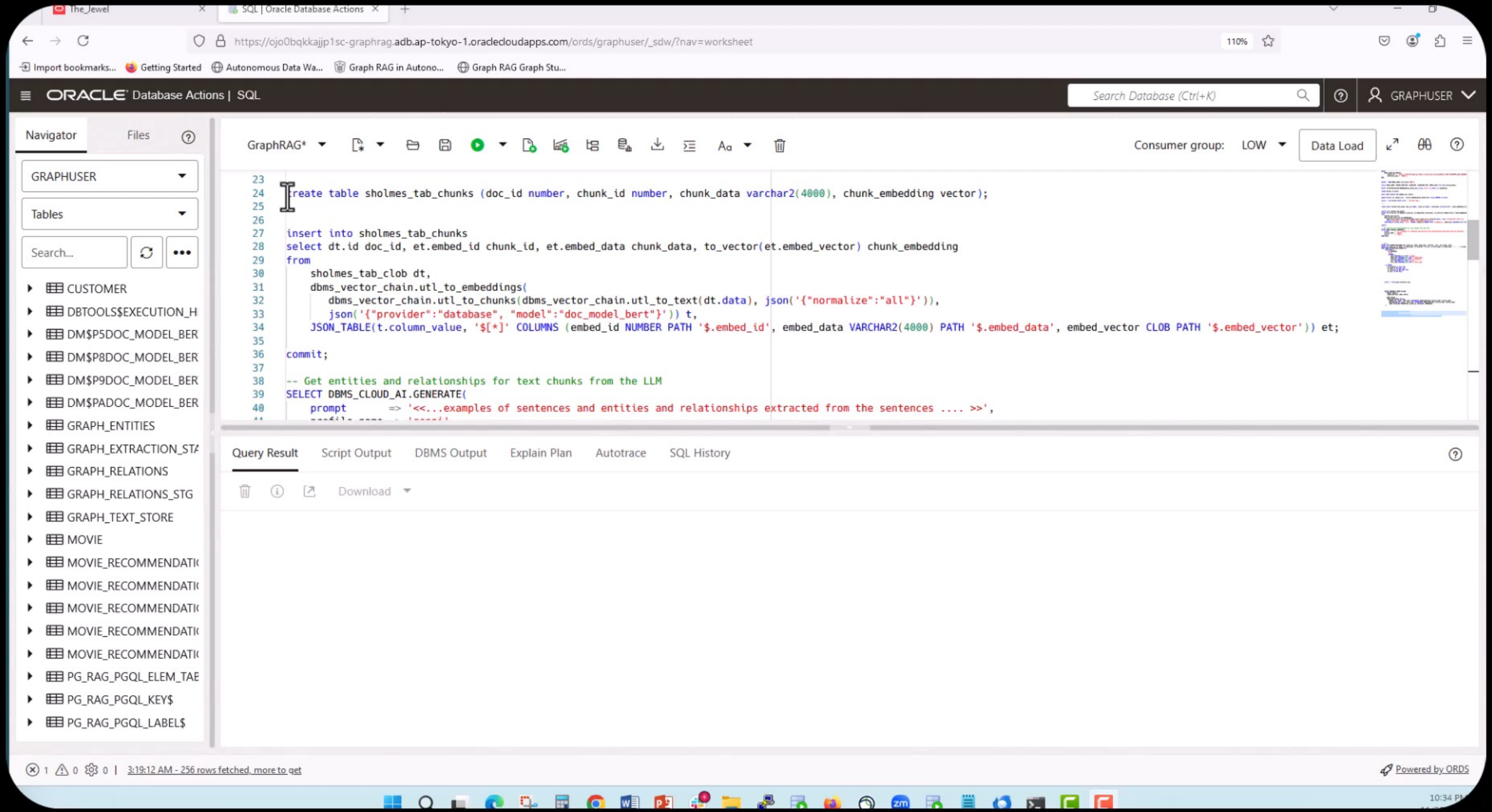
1. Find entities related to **'Who Stole the Jewel?'**
2. Retrieve text chunks associated to those entities

   (**The jewel robbery** -> LOCATION_OF -> Hotel cosmopolitan)

   (Ryder -> EMPLOYED_BY -> Hotel cosmopolitan)

   (Ryder -> RIFLED -> The jewel case)

Answer: James Ryder, the hotel attendant, is implicated in the theft of the jewel.


The_Jewel
SQL | Oracle Database Actions
https://ojo0bqkkajjp1sc-graphrag.adb.ap-tokyo-1.oraclecloudapps.com/ords/graphuser/_sdw/?nav=worksheet
110%
Import bookmarks...
Getting Started
Autonomous Data Wa...
Graph RAG in Autono...
Graph RAG Graph Stu...
ORACLE Database Actions | SQL
Search Database (Ctrl+K)
GRAPHUSER
Navigator
Files
GRAPHUSER
Tables
Search...
CUSTOMER
DBTOOLS$EXECUTION_H
DM$P5DOC_MODEL_BER
DM$P8DOC_MODEL_BER
DM$P9DOC_MODEL_BER
DM$PADOC_MODEL_BER
GRAPH_ENTITIES
GRAPH_EXTRACTION_STA
GRAPH_RELATIONS
GRAPH_RELATIONS_STG
GRAPH_TEXT_STORE
MOVIE
MOVIE_RECOMMENDATI
MOVIE_RECOMMENDATI
MOVIE_RECOMMENDATI
MOVIE_RECOMMENDATI
MOVIE_RECOMMENDATI
PG_RAG_PGQL_ELEM_TAE
PG_RAG_PGQL_KEY$
PG_RAG_PGQL_LABEL$
GraphRAG*
Consumer group:
LOW
Data Load
23
24 create table sholmes_tab_chunks (doc_id number, chunk_id number, chunk_data varchar2(4000), chunk_embedding vector);
25
26
27 insert into sholmes_tab_chunks
28 select dt.id doc_id, et.embed_id chunk_id, et.embed_data chunk_data, to_vector(et.embed_vector) chunk_embedding
29 from
30 sholmes_tab_clob dt,
31 dbms_vector_chain.utl_to_embeddings(
32 dbms_vector_chain.utl_to_chunks(dbms_vector_chain.utl_to_text(dt.data), json('{"normalize":"all"}')),
33 json('{"provider":"database", "model":"doc_model_bert"}')) t,
34 JSON_TABLE(t.column_value, '$[*]' COLUMNS (embed_id NUMBER PATH '$.embed_id', embed_data VARCHAR2(4000) PATH '$.embed_data', embed_vector CLOB PATH '$.embed_vector')) et;
35
36 commit;
37
38 -- Get entities and relationships for text chunks from the LLM
39 SELECT DBMS_CLOUD_AI.GENERATE(
40 prompt => '<<...examples of sentences and entities and relationships extracted from the sentences .... >>',
Query Result
Script Output
DBMS Output
Explain Plan
Autotrace
SQL History
Download
1 0 0 | 3:19:12 AM - 256 rows fetched, more to get
Powered by ORDS

Graph RAG is an active topic of research

Ideas we are exploring

Use `Select AI` for graph queries

- Will enable translation of natural language queries into SQL/PGQ standard syntax

Options to automate extraction of entities and relationships from text data

Options for entity resolution of extracted entities

Graphs are good at capturing complex interactions of the real world

- **Enable enterprises to use graphs to provide an LLM facts about its business**

Key Takeaways

Graphs provide new insights by helping you capture and navigate relationships in data

New SQL syntax in Oracle Database 23ai makes it easy to write queries to query data based on these connections

Graphs add value to AI by enabling algorithms to be grounded in facts

- Graphs provide a natural way to model complex real-world data

Resources


oracle.com/database/graph/


oracle.com/livelabs search for ‘graph’


YouTube search for ‘oracle spatial and graph’


blogs.oracle.com/database/category/db-graph


medium.com/tag/oracle-graph/latest

Thank you

linkedin.com/in/melliyal
Melliyal.Annamalai@oracle.com


ORACLE

Q&A Open

Important links to bookmark

Links to get you started and to keep up to date with Autonomous Database

1 New Get Started page:
oracle.com/autonomous-database/get-started/

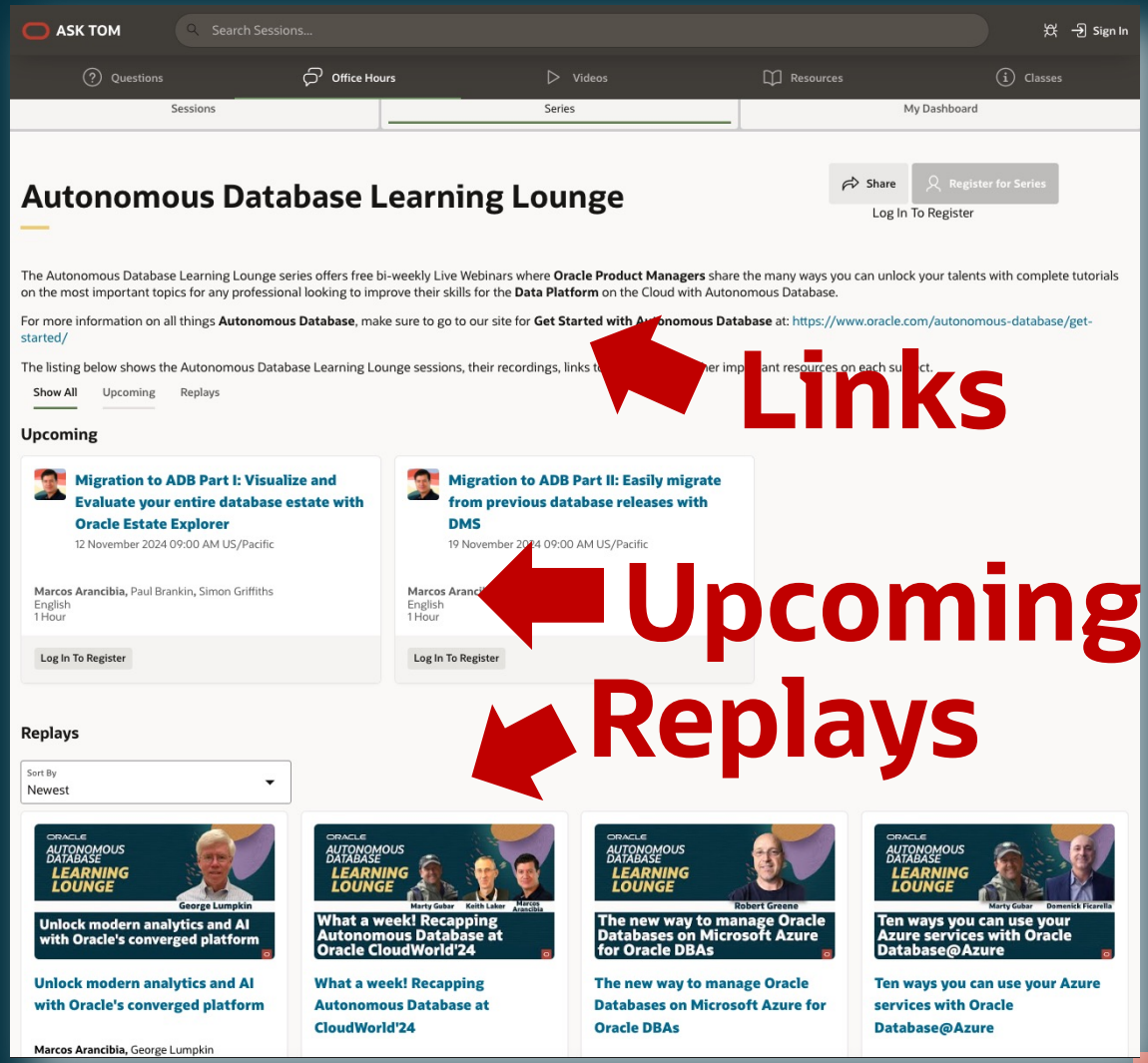
2 Join us: LinkedIn bit.ly/adb-linkedin-grp @AutonomousDW

3 Got a question?
We are on stackoverflow
bit.ly/adb-stackoverflow

Join us on Developers Slack
(search #oracle-autonomous-database)
bit.ly/odevrel_slack (odevrel_slack)

Final Thoughts

oracle.com/goto/adb-learning-lounge



Thank you for joining !!!

AUTONOMOUS DATABASE

LEARNING LOUNGE