Graphs, Graph-RAG, and Generative AI: An Introduction

Hans Viehmann, Oracle

Heli Helskyaho (@helifromfinland),Miracle Finland Oy

MIRACLE

Miracle Finland Oy

Hans

- Product Manager for Oracle Spatial and Graph
- Studied physics in Hamburg Univ. and Imperial College, London
- Working in IT since 1986, started with Oracle7 in 1993
- Joined Oracle in 1994, specialized in Oracle Spatial in 1995

MIRACLE
Miracle Finland Oy

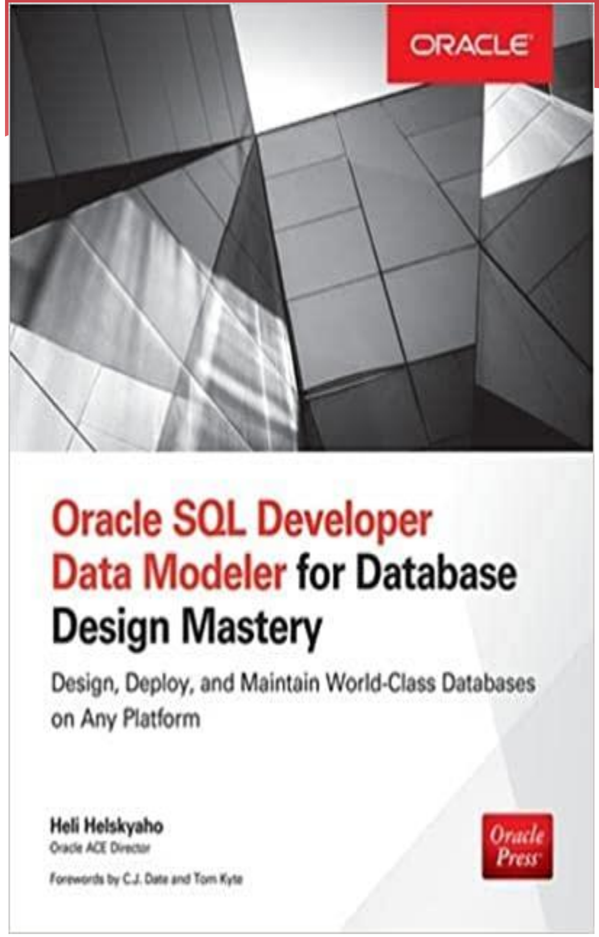
Heli

- Graduated from University of Helsinki (Master of Science, computer science), currently a doctoral student, researcher and lecturer at University of Helsinki
- Worked with Oracle products since 1993, worked for IT since 1990
- Data and Database!
- CEO for Miracle Finland Oy
- Oracle ACE Director
- Public speaker and an author
- Author of the book Oracle SQL Developer Data Modeler for Database Design Mastery (Oracle Press, 2015), co-author for Real World SQL and PL/SQL: Advice from the Experts (Oracle Press, 2016), Machine Learning for Oracle Database Professionals: Deploying Model-Driven Applications and Automation Pipelines (Apress, 2021), and Extending Oracle Application Express with Oracle Cloud Features: A Guide to Enhancing APEX Web Applications with Cloud-Native and Machine Learning Technologies (Apress, 2022)

A Oracle ACE Director

MiRACLE
Miracle Finland Oy

Books



ORACLE

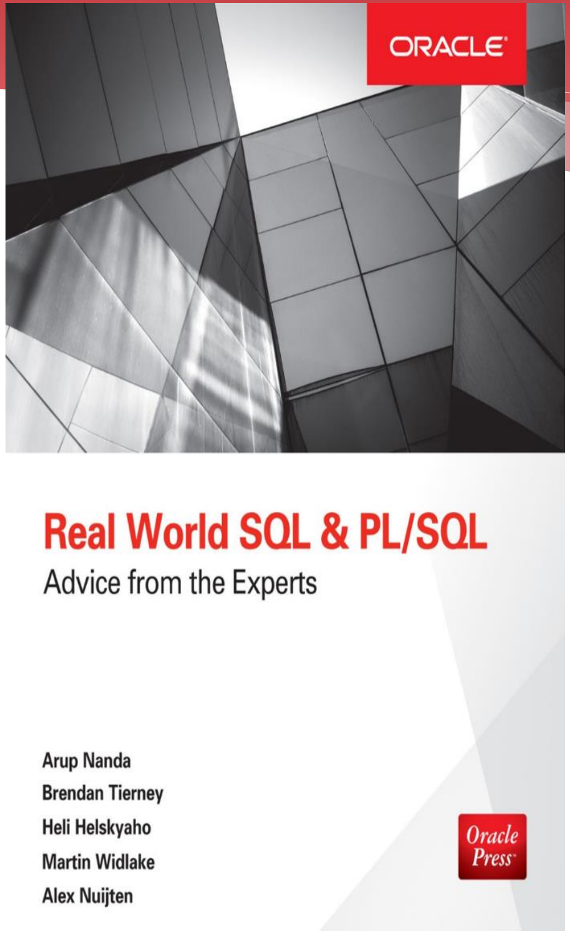
Oracle SQL Developer Data Modeler for Database Design Mastery

Design, Deploy, and Maintain World-Class Databases on Any Platform

Heli Helskyaho
Oracle ACE Director

Forewords by C.J. Date and Tom Kyte

Oracle Press



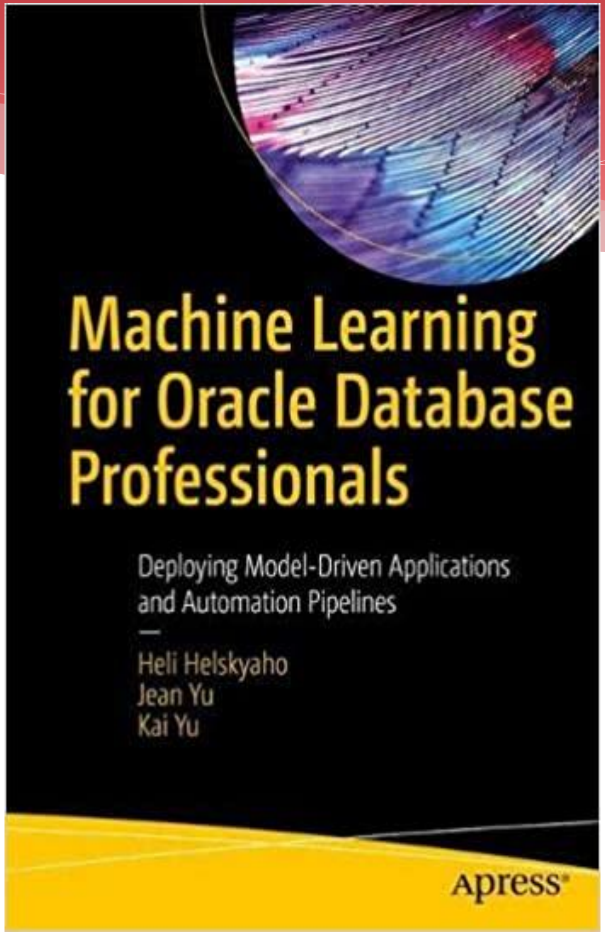
ORACLE

Real World SQL & PL/SQL

Advice from the Experts

Arup Nanda
Brendan Tierney
Heli Helskyaho
Martin Widlake
Alex Nuijten

Oracle Press



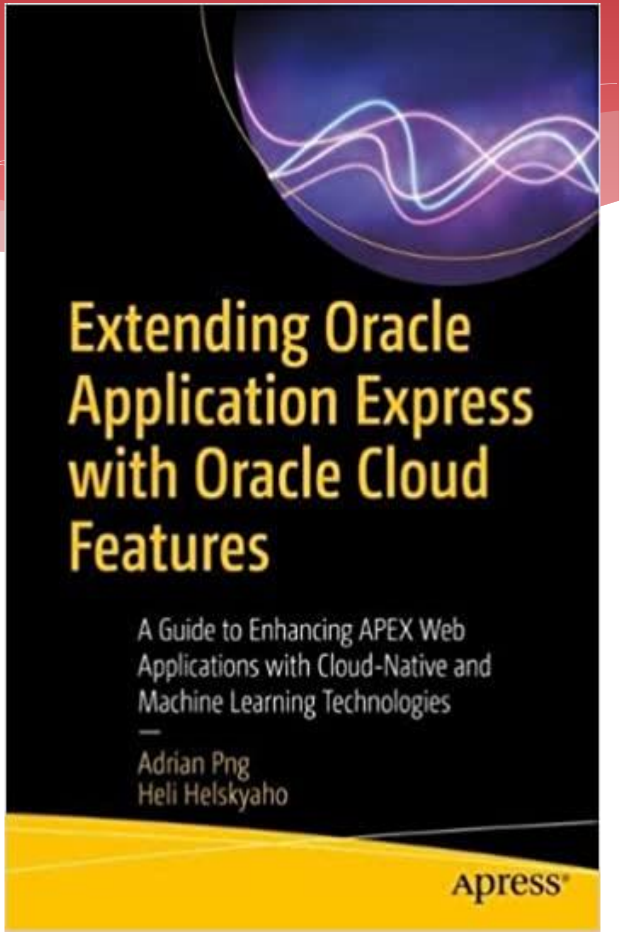
Machine Learning for Oracle Database Professionals

Deploying Model-Driven Applications and Automation Pipelines

—

Heli Helskyaho
Jean Yu
Kai Yu

Apress®



Extending Oracle Application Express with Oracle Cloud Features

A Guide to Enhancing APEX Web Applications with Cloud-Native and Machine Learning Technologies

—

Adrian Png
Heli Helskyaho

Apress®

MIRACLE
Miracle Finland Oy

Oracle ACE

The Oracle ACE Program

400+ technical experts helping peers globally

- The Oracle ACE Program recognizes and rewards community members for their technical and community contributions to the Oracle community
- 3 membership levels: Director, Pro, and Associate
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com

✉ aceprogram_ww@oracle.com   Facebook.com/OracleACEs   X @oracleace   in Oracle ACE Program Group

ACE Member Benefits

Key Benefits

Cool **swag***, Digital awards for social media, Oracle **CloudWorld pass***, & more

Direct Access to Product Management

Multiple **direct communication channels** to product management and fellow ACEs

Exclusive Content

Exclusive **monthly virtual meetings** with product development teams + engaging guest speakers

Networking

In-person & virtual networking opportunities for ACEs to **connect with product development** and each other.

Cloud Account

$5k USD Cloud account*

Travel Support

ACE Directors are eligible for travel support to give presentations or lead workshops at conferences globally

* for Pro and Director levels

Why would you use Graph Databases?

Graphs are useful when **discovering** and understanding relationships is important

For example, with this data…

Customers

cst_id	name
c1	Alice
c2	Bob
c3	Charlie
c4	Dave
…	
…	

Accounts

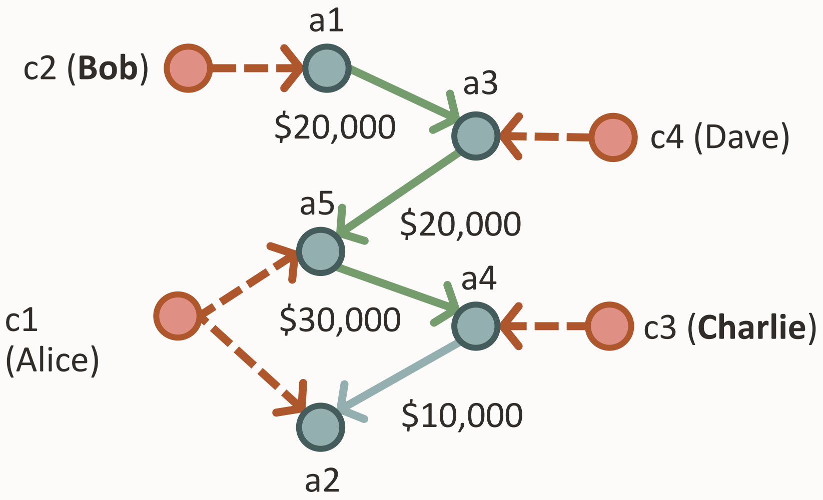
acc_id	cst_id
a1	c2
a2	c1
a3	c4
a4	c3
a5	c1
…	

Transactions

src_acc	dst_acc	amount	date
a1	a3	$20,000	2020-10-01
a5	a4	$30,000	2020-10-02
a4	a2	$10,000	2020-10-03
a3	a5	$20,000	2020-10-04
…			
…			

Why would you use Graph Databases?

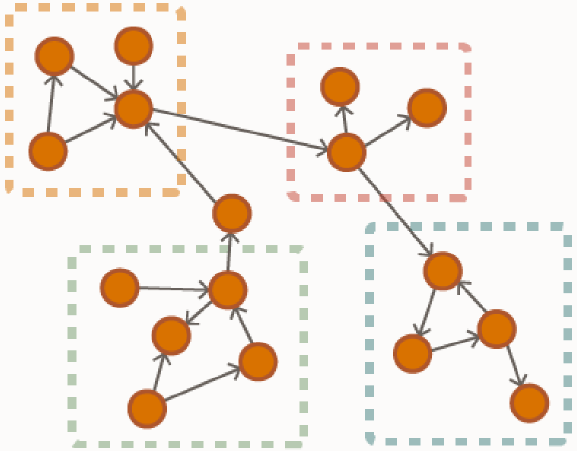
*Is there **any money flow** between Bob & Charlie?*



Graph-based queries:

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

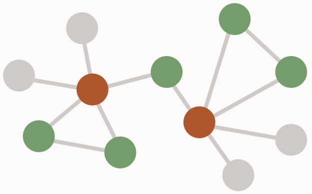
*How to identify the **clusters** of transactions?*



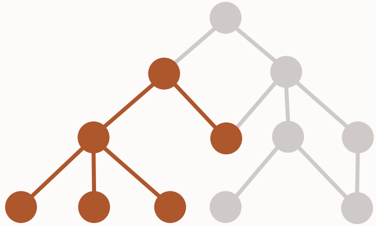
Graph algorithms:

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

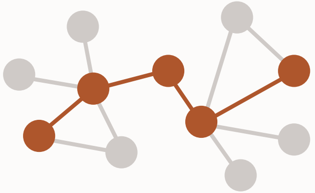
Analyzing relationships, direct or indirect



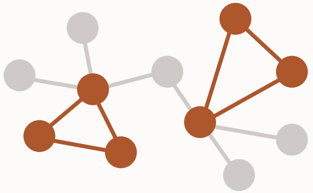
Networks



Hierarchies



Paths



Patterns

Use case: Creating Knowledge Graphs from Documents

Decompose document into chunks

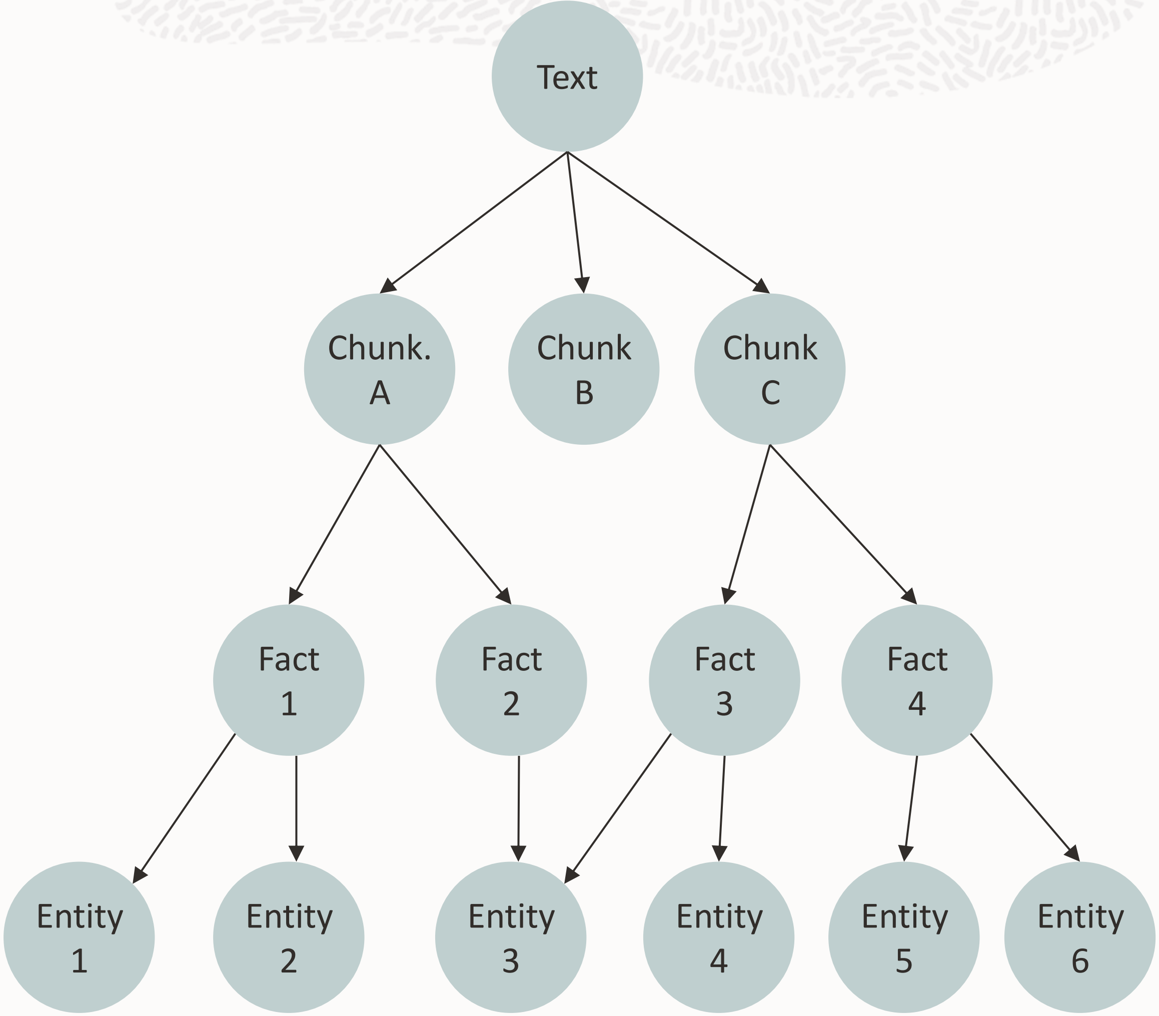
- Enables computation of embeddings

Capture relevant information

- Extract entities
- Extract facts (entities and relationships)
- Resolve co-references

Benefits

- Compact representation enabling similarity search
- Links facts back to their origin
- Helpful for explainability, governance, compliance



Oracle Database has comprehensive support for creating, querying, managing, analyzing graphs

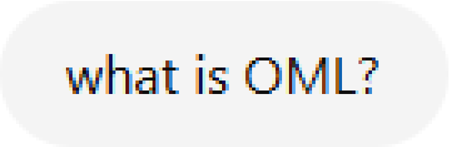
Generative AI

Generative AI, Gartner predictions (April 12, 2024)

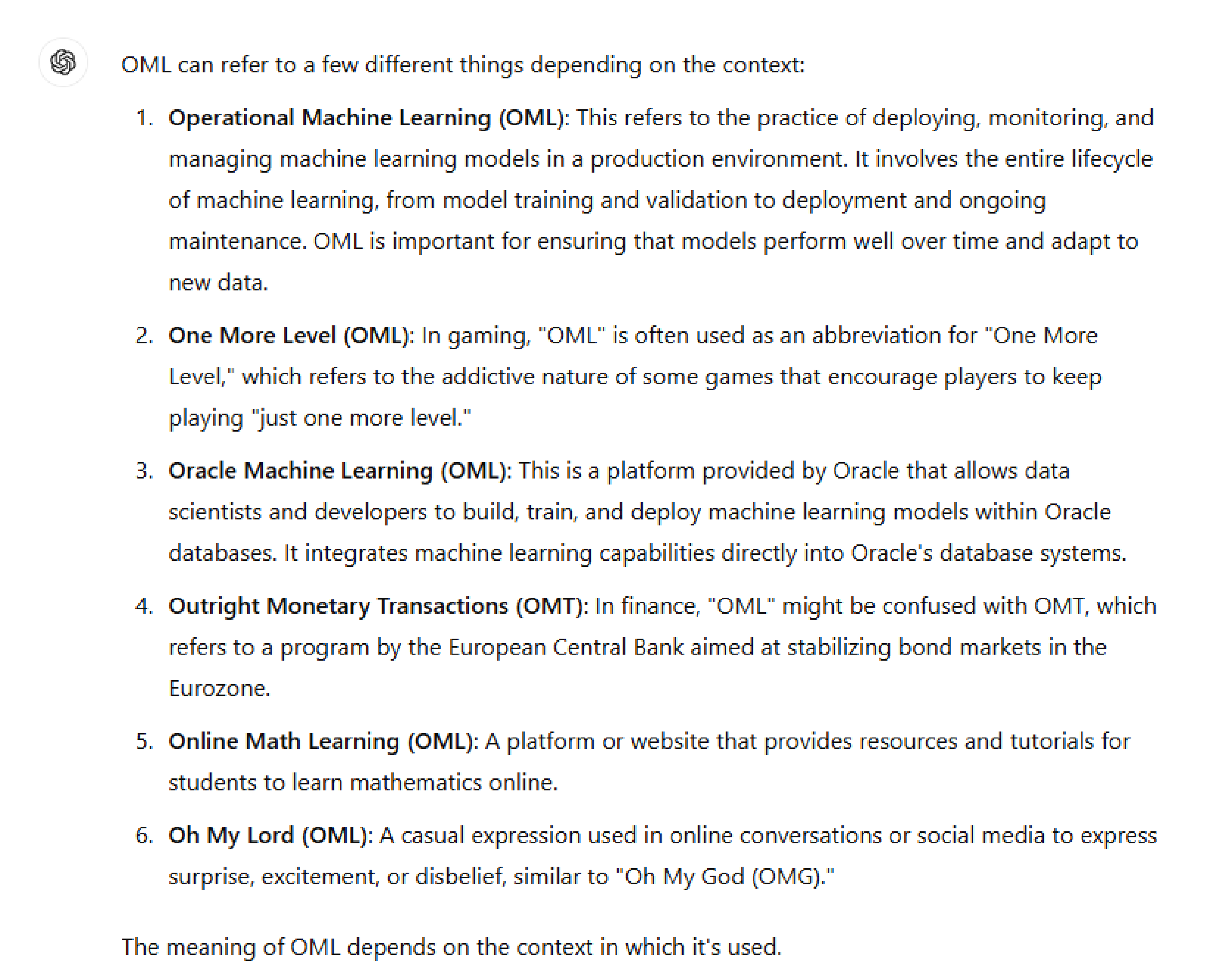
- “**Generative AI technologies** will greatly evolve in the next four years.”
- “With continued investment from large technology and research companies alongside enterprising startups, **GenAI will continue to make huge strides in the coming year.**”
- “Gartner predictions — and our recommendations for acting on this foresight — **equip IT leaders to stay ahead.”**

https://www.gartner.com/en/articles/3-bold-and-actionable-predictions-for-the-future-of-genai

Generative AI without the context (ChatGPT)



what is OML?



OML can refer to a few different things depending on the context:

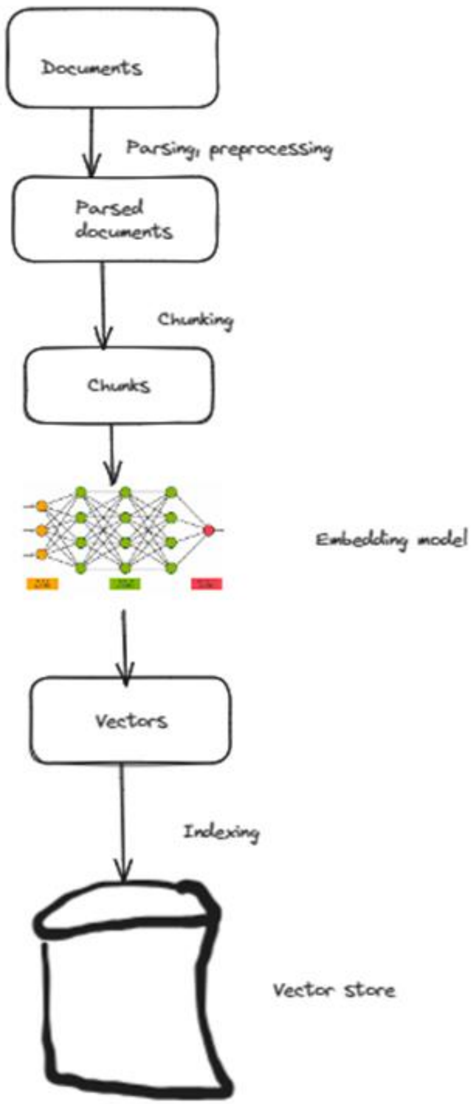
1. **Operational Machine Learning (OML)**: This refers to the practice of deploying, monitoring, and managing machine learning models in a production environment. It involves the entire lifecycle of machine learning, from model training and validation to deployment and ongoing maintenance. OML is important for ensuring that models perform well over time and adapt to new data.
2. **One More Level (OML)**: In gaming, "OML" is often used as an abbreviation for "One More Level," which refers to the addictive nature of some games that encourage players to keep playing "just one more level."
3. **Oracle Machine Learning (OML)**: This is a platform provided by Oracle that allows data scientists and developers to build, train, and deploy machine learning models within Oracle databases. It integrates machine learning capabilities directly into Oracle's database systems.
4. **Outright Monetary Transactions (OMT)**: In finance, "OML" might be confused with OMT, which refers to a program by the European Central Bank aimed at stabilizing bond markets in the Eurozone.
5. **Online Math Learning (OML)**: A platform or website that provides resources and tutorials for students to learn mathematics online.
6. **Oh My Lord (OML)**: A casual expression used in online conversations or social media to express surprise, excitement, or disbelief, similar to "Oh My God (OMG)."

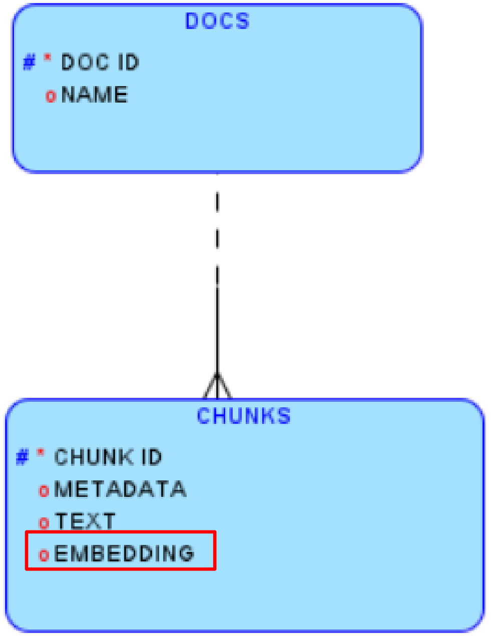
The meaning of OML depends on the context in which it's used.

Retrieval-Augmented Generation, RAG

- Adds the context
- Augments the LLM with your own data
- Grounds the responses to your data
- Reduces hallucination

RAG example



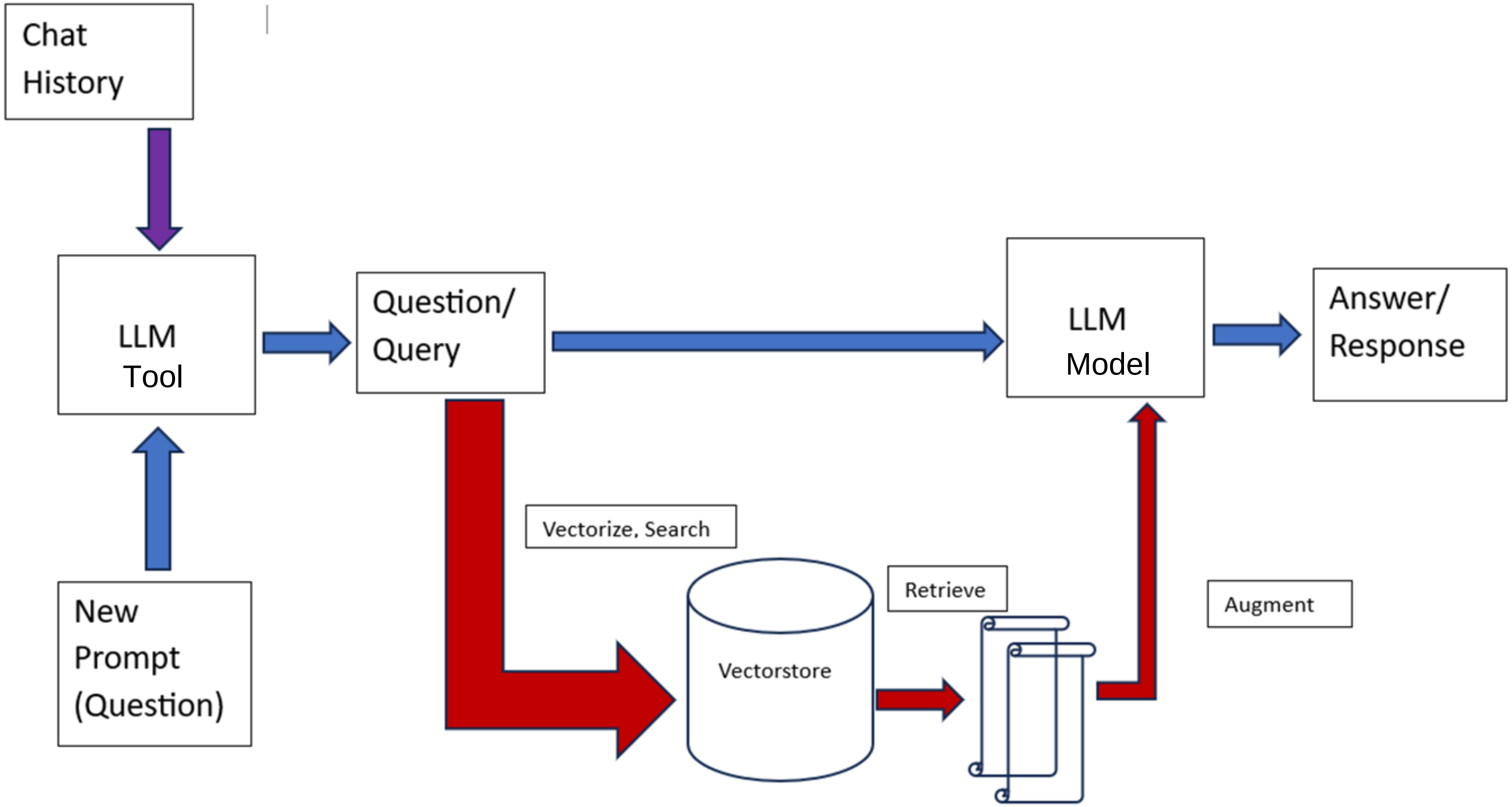

DOCS

# * DOC ID
o NAME

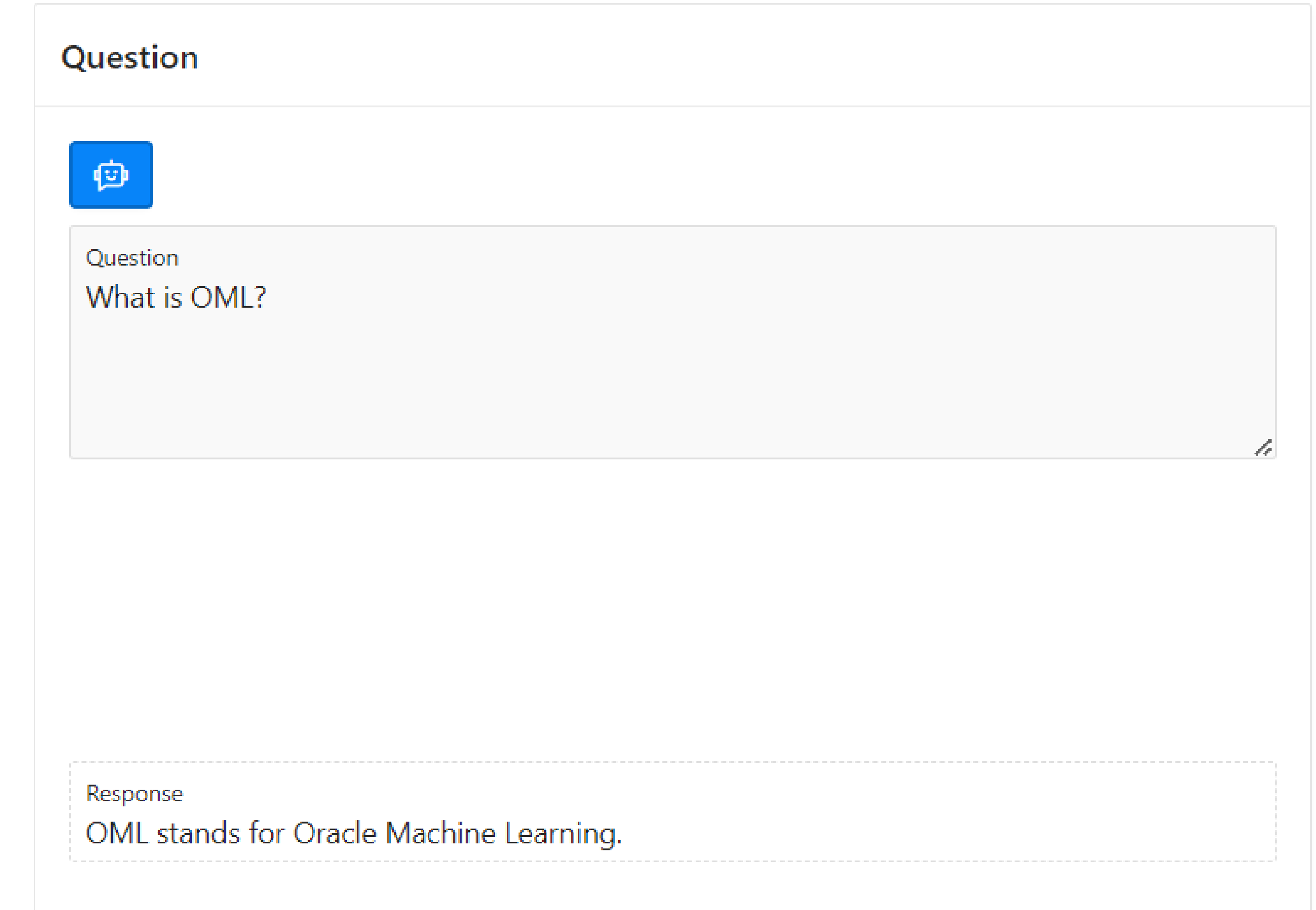
CHUNKS

# * CHUNK ID
o METADATA
o TEXT
o EMBEDDING

MIRACLE
Miracle Finland Oy


Chat History
LLM Tool
New Prompt (Question)
Question/ Query
Vectorize, Search
Vectorstore
Retrieve
Augment
LLM Model
Answer/ Response

RAG solution



Question

Question

What is OML?

Response

OML stands for Oracle Machine Learning.

A graph…

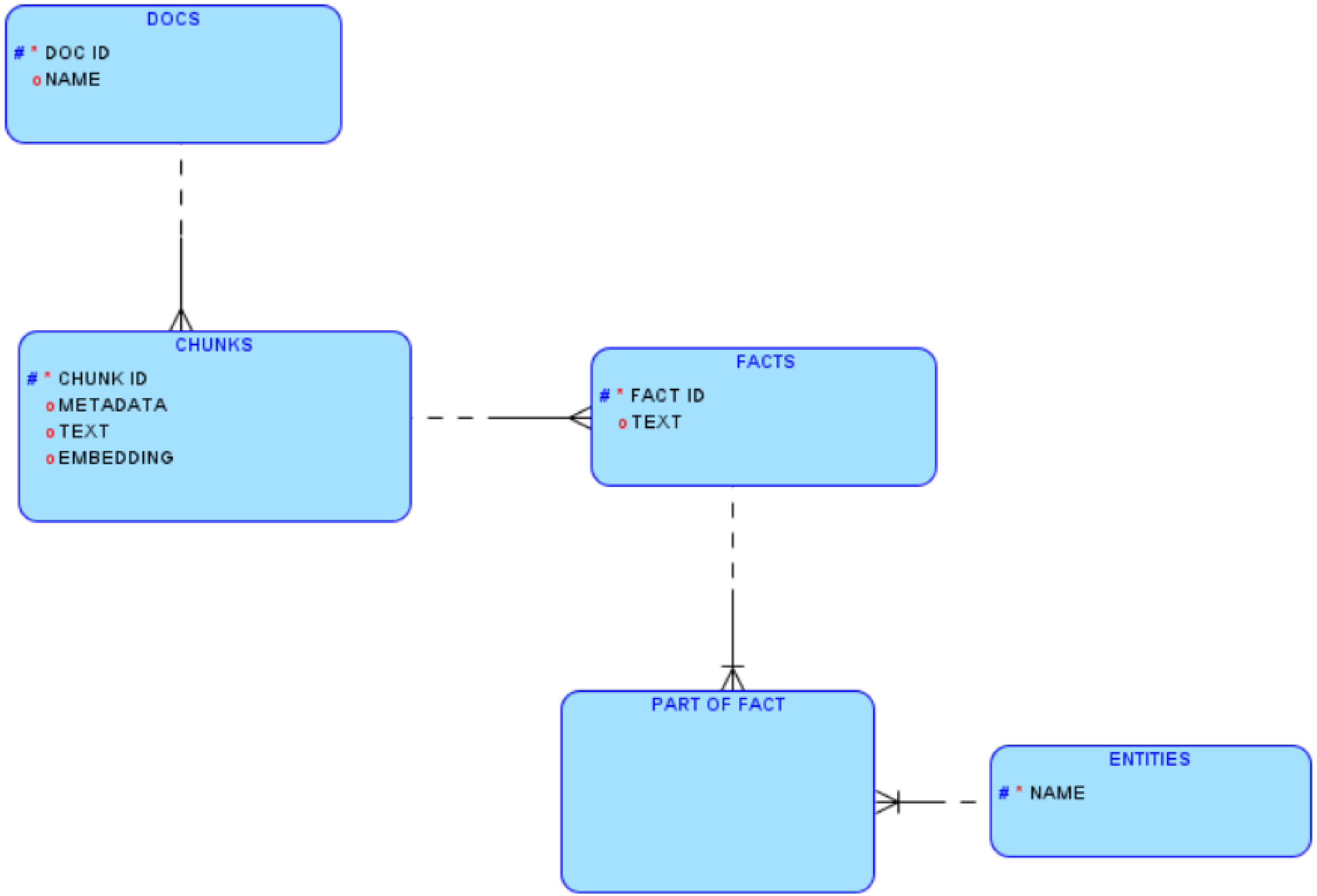
- …is all about relationships.
- Think about a document
  - A sentence
  - A paragraph
  - A page
  - A document
  - … they are all about the relationships of words.

How about…

- … we combine the RAG and the Graph?

GraphRAG

- Key elements (entities)
- Atomic facts
- … Of a chunk
- A knowledge graph captures both the semantics and the context


DOCS
# * DOC ID
o NAME
CHUNKS
# * CHUNK ID
o METADATA
o TEXT
o EMBEDDING
FACTS
# * FACT ID
o TEXT
PART OF FACT
ENTITIES
# * NAME

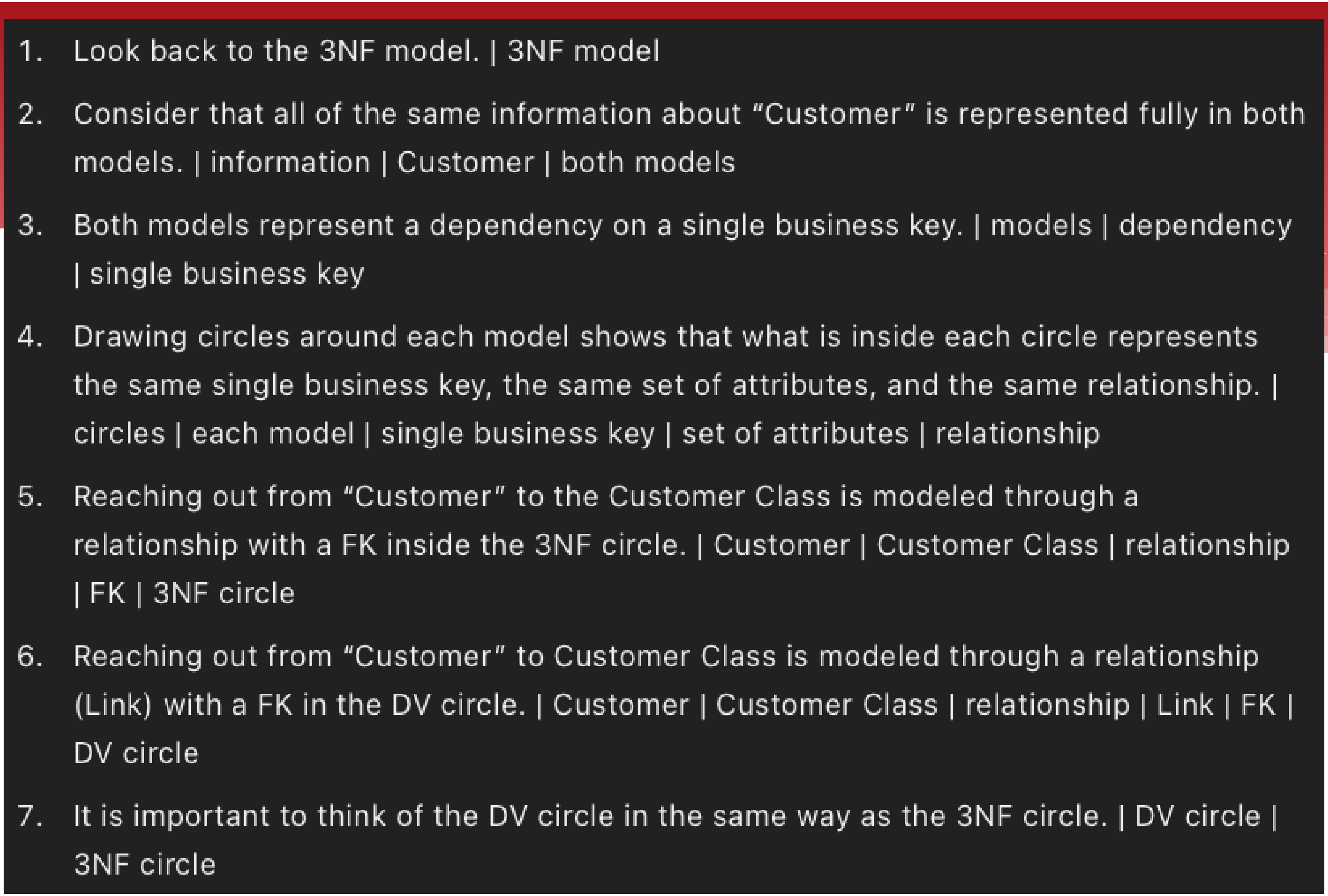
An example from a Data Vault 2.0 documentation

Prompt LLM for key elements (entities) and atomic facts

You are now an intelligent assistant tasked with meticulously extracting both key elements and atomic facts from a long text.

1. Key Elements: The essential nouns (e.g., characters, times, events, places, numbers), verbs (e.g., actions), and adjectives (e.g., states, feelings) that are pivotal to the text's narrative.

2. Atomic Facts: The smallest, indivisible facts, presented as concise sentences. These include propositions, theories, existences, concepts, and implicit elements like logic, causality, event sequences, interpersonal relationships, timelines, etc...



1. Look back to the 3NF model. | 3NF model
2. Consider that all of the same information about “Customer” is represented fully in both models. | information | Customer | both models
3. Both models represent a dependency on a single business key. | models | dependency | single business key
4. Drawing circles around each model shows that what is inside each circle represents the same single business key, the same set of attributes, and the same relationship. | circles | each model | single business key | set of attributes | relationship
5. Reaching out from “Customer” to the Customer Class is modeled through a relationship with a FK inside the 3NF circle. | Customer | Customer Class | relationship | FK | 3NF circle
6. Reaching out from “Customer” to Customer Class is modeled through a relationship (Link) with a FK in the DV circle. | Customer | Customer Class | relationship | Link | FK | DV circle
7. It is important to think of the DV circle in the same way as the 3NF circle. | DV circle | 3NF circle

Prompt to unify the key elements

You are now an intelligent assistant tasked with standardizing key words in a json like structure.

Requirements:

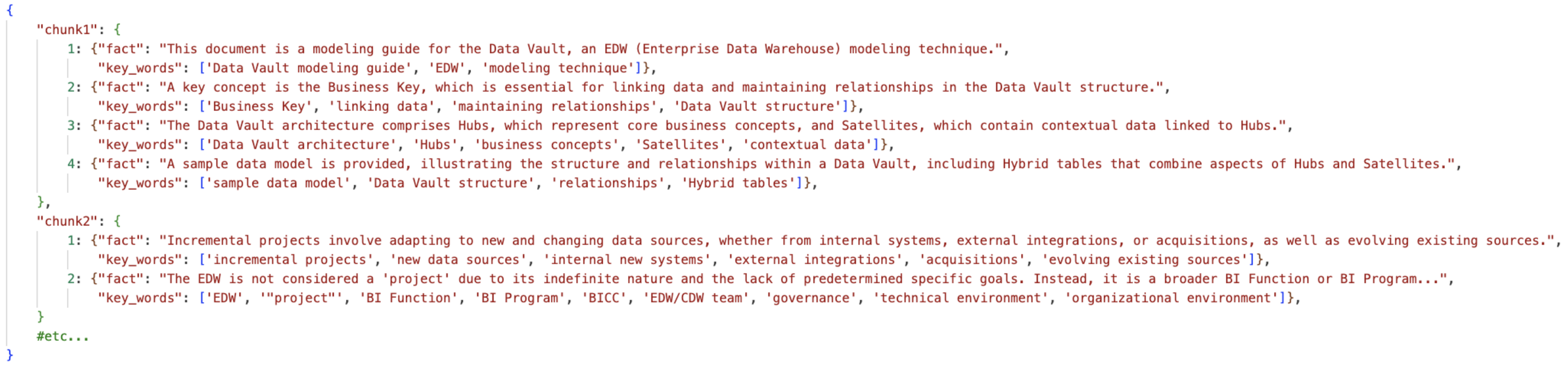
#####

1. Ensure that all key words that existed still exist in your version

2. Similar keywords should be standardized so as to remove unnecessary duplicates

etc…

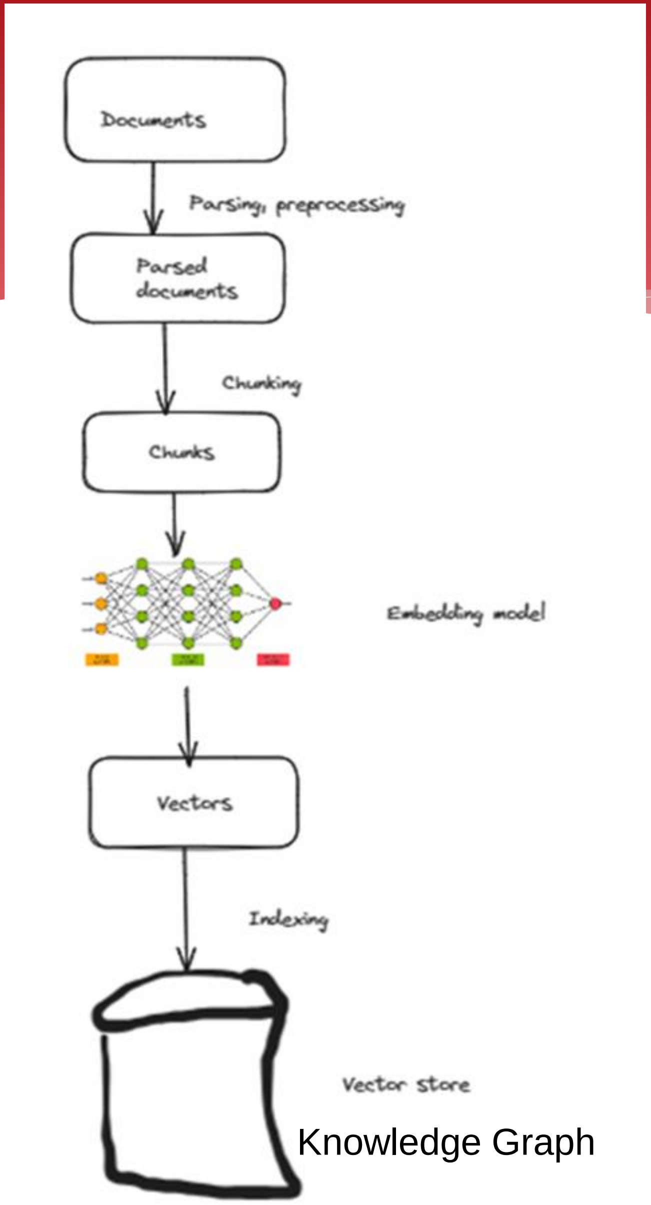
Chunks with their facts and key elements



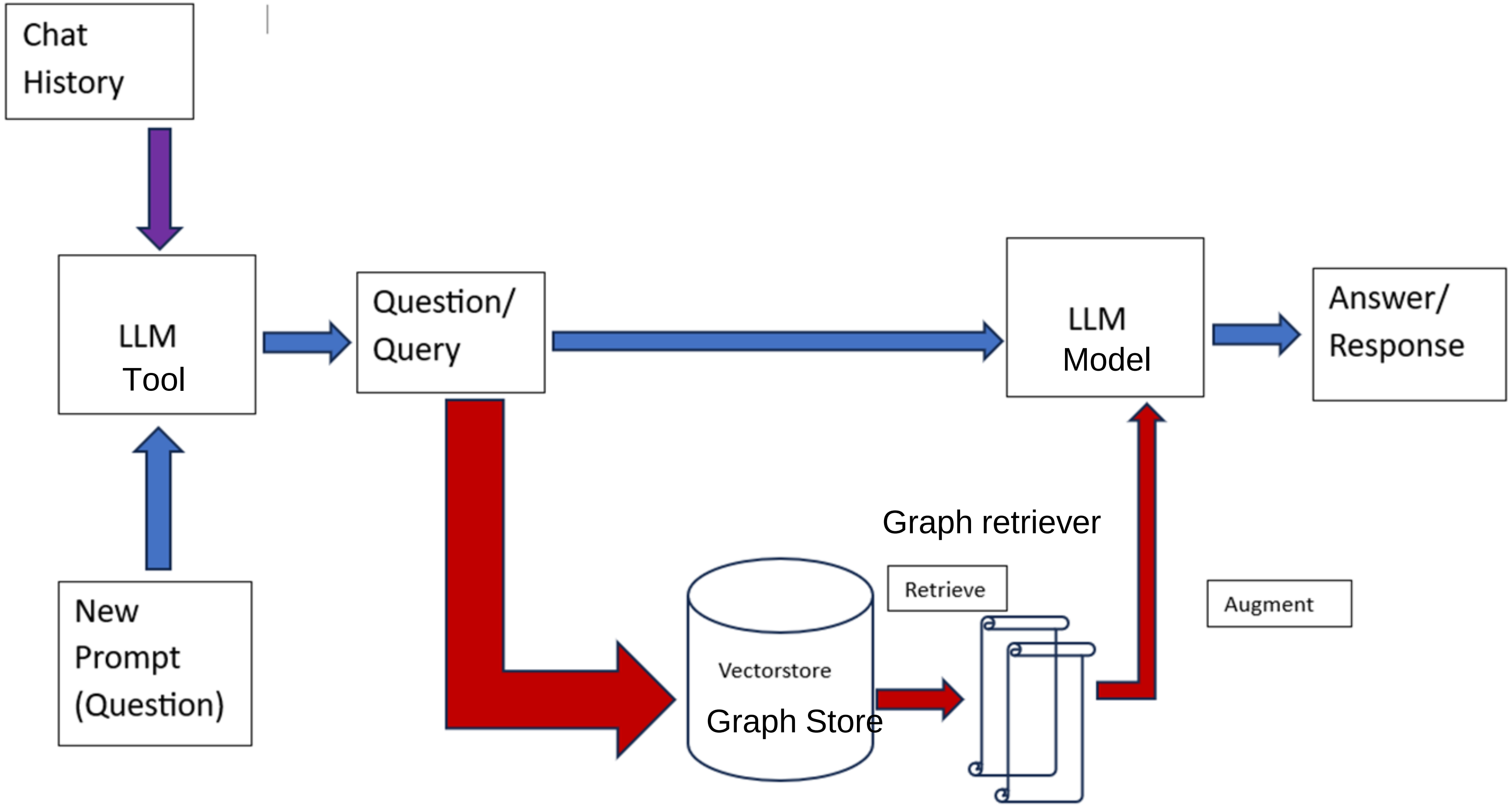
```
{
    "chunk1": {
        1: {"fact": "This document is a modeling guide for the Data Vault, an EDW (Enterprise Data Warehouse) modeling technique.",
            "key_words": ['Data Vault modeling guide', 'EDW', 'modeling technique']},
        2: {"fact": "A key concept is the Business Key, which is essential for linking data and maintaining relationships in the Data Vault structure.",
            "key_words": ['Business Key', 'linking data', 'maintaining relationships', 'Data Vault structure']},
        3: {"fact": "The Data Vault architecture comprises Hubs, which represent core business concepts, and Satellites, which contain contextual data linked to Hubs.",
            "key_words": ['Data Vault architecture', 'Hubs', 'business concepts', 'Satellites', 'contextual data']},
        4: {"fact": "A sample data model is provided, illustrating the structure and relationships within a Data Vault, including Hybrid tables that combine aspects of Hubs and Satellites.",
            "key_words": ['sample data model', 'Data Vault structure', 'relationships', 'Hybrid tables']},
    },
    "chunk2": {
        1: {"fact": "Incremental projects involve adapting to new and changing data sources, whether from internal systems, external integrations, or acquisitions, as well as evolving existing sources.",
            "key_words": ['incremental projects', 'new data sources', 'internal new systems', 'external integrations', 'acquisitions', 'evolving existing sources']},
        2: {"fact": "The EDW is not considered a 'project' due to its indefinite nature and the lack of predetermined specific goals. Instead, it is a broader BI Function or BI Program...",
            "key_words": ['EDW', '"project"', 'BI Function', 'BI Program', 'BICC', 'EDW/CDW team', 'governance', 'technical environment', 'organizational environment']},
    }
    #etc...
}
```

Save the data to the Oracle Database in tables designed earlier

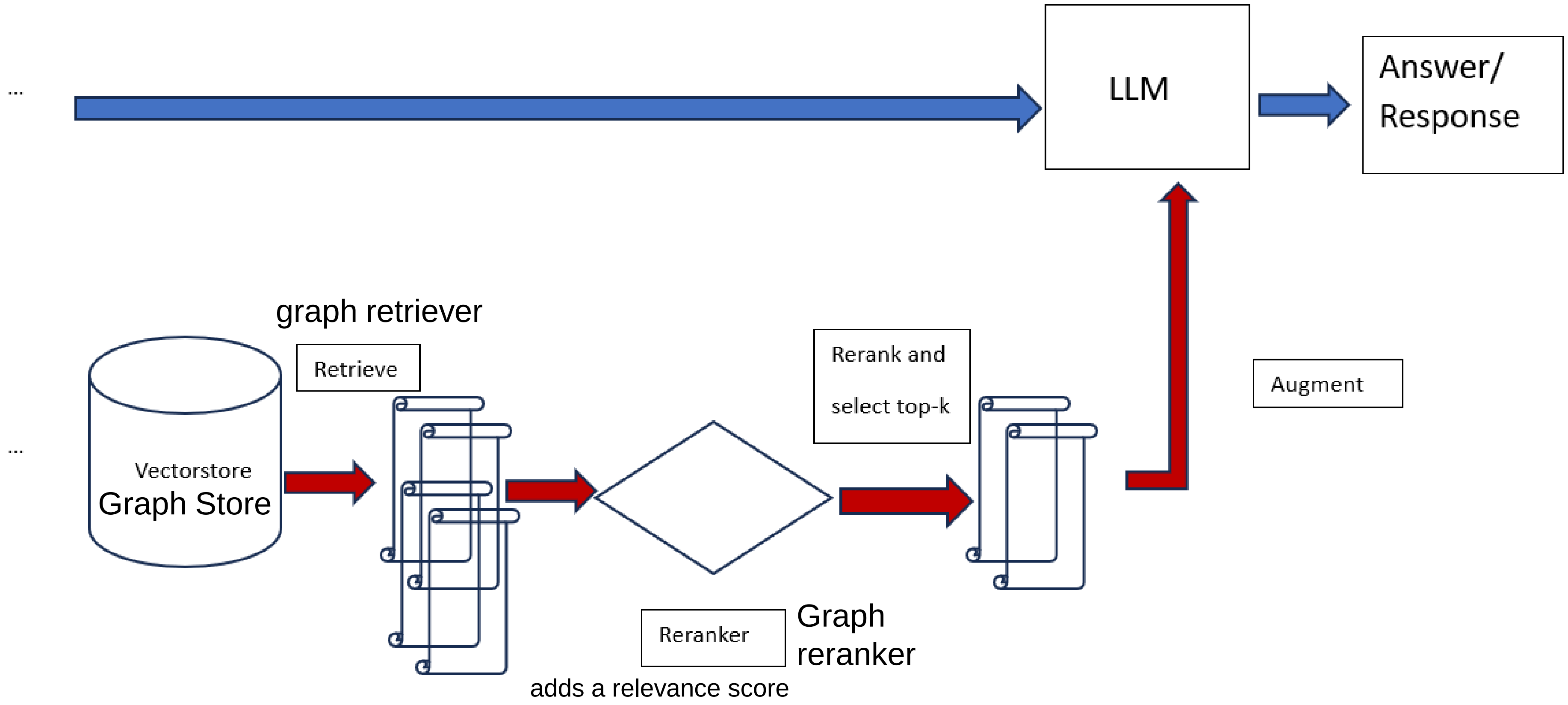
Create a graph manually (you can ask LLM to give the script)



Knowledge Graph


Chat History
LLM Tool
New Prompt (Question)
Question/ Query
LLM Model
Answer/ Response
Graph retriever
Retrieve
Augment
Vectorstore
Graph Store

Reranking



Copyright © Miracle Finland Oy

Why is it better with graph?

- ”Things, not strings”*
- You will not miss important content as might happen with a regular RAG solution
- If the graph has been build correctly (remember the phase of unifying?) we will avoid duplicates
- Higher accuracy, more complete answers (useful, relevant)
- Better explainability, traceability

*https://blog.google/products/search/introducing-knowledge-graph-things-not/

Multi-hop reasoning

- a single question broken down into multiple sub-questions
- getting an accurate answer requires retrieval of numerous documents
- Getting a correct (or at least useful) answer might be impossible

Multi-hop question-answering task

- Did any of the former OpenAI employees publish papers about AGI?
- This question is multi-part containing two questions:
  - Who are the former employees of OpenAI?
  - Did any of them publish a paper about AGI?
- Chain-of-Thought: who has worked for OpenAI? Who of those have published a paper about AGI?
  - Not very user friendly, multiple LLM calls
- With a graph?
  - Simple, graph can follow chains of relationships, facilitating more complex reasoning

Benchmark?

- ”A Knowledge Graph improved LLM response accuracy by 3x across 43 business questions.”
- “LLMs – without the support of a Knowledge Graph – fail to accurately answer “schema-intensive” questions (questions often focused on metrics & KPIs and strategic planning). LLMs returned accurate responses 0% of the time.”
- “A Knowledge Graph significantly improves the accuracy of LLM responses – even schema-intensive questions.”

https://data.world/blog/generative-ai-benchmark-increasing-the-accuracy-of-llms-in-the-enterprise-with-a-knowledge-graph/

LangChain

- or LlamaIndex
  - KnowledgeGraphIndex
  - PropertyGraphIndex
- or… any other libraries for building search and retrieval applications
- or.. PL/SQL

A document to graph using LangChain

```
# Get LLM from OCI GenAI
llm = ChatOCIGenAI(
    model_id="meta.llama-3-70b-instruct",
    service_endpoint="https://inference.generativeai.eu-frankfurt-1.oci.oraclecloud.com",
    compartment_id=COMPARTMENT_OCID,
    model_kwargs={"temperature": 0.2, "max_tokens": 2000},
  )

# Initialize the graph transformer
llm_transformer = LLMGraphTransformer(llm=llm)

# Transform text to graph
documents = [Document(page_content=text)]
graph_documents = llm_transformer.convert_to_graph_documents(documents)
```

Create the objects in a database

```
CREATE TABLE NODES(
  NODE_ID VARCHAR2(255) PRIMARY KEY,
  TYPE VARCHAR2(255));

CREATE TABLE RELATIONSHIPS(
  SOURCE_NODE VARCHAR2(255) CONSTRAINT fk1 REFERENCES NODES(NODE_ID),
  DESTINATION_NODE VARCHAR2(255) CONSTRAINT fk2 REFERENCES NODES(NODE_ID),
  RELATION_TYPE VARCHAR2(255),
  CONSTRAINT pk PRIMARY KEY(SOURCE_NODE, DESTINATION_NODE));

CREATE PROPERTY GRAPH node_graph
 VERTEX TABLES (
  NODES KEY (NODE_ID)
    LABEL node_info
      PROPERTIES (NODE_ID, TYPE)
 )
 EDGE TABLES (
  RELATIONSHIPS
    SOURCE KEY (source_node) REFERENCES NODES(NODE_ID)
    DESTINATION KEY (destination_node) REFERENCES NODES(NODE_ID)
    PROPERTIES (RELATION_TYPE)  );
```

MIRACLE
Miracle Finland Oy

Print nodes and relationships

```
# Print nodes and relationships
print(f"Nodes:{graph_documents[0].nodes}")
print(f"Relationships:{graph_documents[0].relationships}")

# Outputs:
nodes = [Node(id='Data Vault', type='Program'), Node(id='natural business relationships', type='Relationship'), Node(id='Core Business Concept', type='Concept'),
Node(id='manages time slice data', type='Characteristic'), Node(id='no descriptive information', type='Characteristic'), Node(id='no FKs', type='Characteristic'),
Node(id='descriptive information', type='Information'), Node(id='Satellite', type='Component'), Node(id='Hub', type='Component'), Node(id='Link',
type='Component')]
relationships = [Relationship(source=Node(id='Hub', type='Component'), target=Node(id='Core Business Concept', type='Concept'), type='REPRESENTS'),
Relationship(source=Node(id='Hub', type='Component'), target=Node(id='no descriptive information', type='Characteristic'), type='HAS_CHARACTERISTIC'),
Relationship(source=Node(id='Hub', type='Component'), target=Node(id='no FKs', type='Characteristic'), type='HAS_CHARACTERISTIC'),
Relationship(source=Node(id='Link', type='Component'), target=Node(id='natural business relationships', type='Relationship'), type='REPRESENTS'),
Relationship(source=Node(id='Link', type='Component'), target=Node(id='no descriptive information', type='Characteristic'), type='HAS_CHARACTERISTIC'),
Relationship(source=Node(id='Satellite', type='Component'), target=Node(id='descriptive information', type='Information'), type='CONTAINS'),
Relationship(source=Node(id='Satellite', type='Component'), target=Node(id='manages time slice data', type='Characteristic'), type='HAS_CHARACTERISTIC'),
Relationship(source=Node(id='Data Vault', type='Program'), target=Node(id='Hub', type='Component'), type='CONSISTS_OF'), Relationship(source=Node(id='Data
Vault', type='Program'), target=Node(id='Link', type='Component'), type='CONSISTS_OF'), Relationship(source=Node(id='Data Vault', type='Program'),
target=Node(id='Satellite', type='Component'), type='CONSISTS_OF')]
```

Insert the data to an Oracle Database

```
# Connect to db
with oracledb.connect(user=USER, password=PWD, dsn=DSN, wallet_location=WLOC, wallet_password=WPWD) as connection:
  # Define insertion queries
  node_query = "INSERT INTO NODES VALUES (:1, :2)"
  relationships_query = "INSERT INTO RELATIONSHIPS VALUES (:1, :2, :3)”

  cursor = connection.cursor()

    # Insert nodes
  for node in nodes:
    cursor.execute(node_query, [node.id, node.type])

  connection.commit()

    # Insert relationships
  for r in relationships:
    cursor.execute(relationships_query, [r.source.id, r.target.id, r.type])

  connection.commit()

# PGQL-query to get the whole graph
SELECT * FROM MATCH (n1:NODES) -[r:RELATIONSHIPS]-> (n2:NODES) ON graph_rag.node_graph
```

Part of the document given to the process

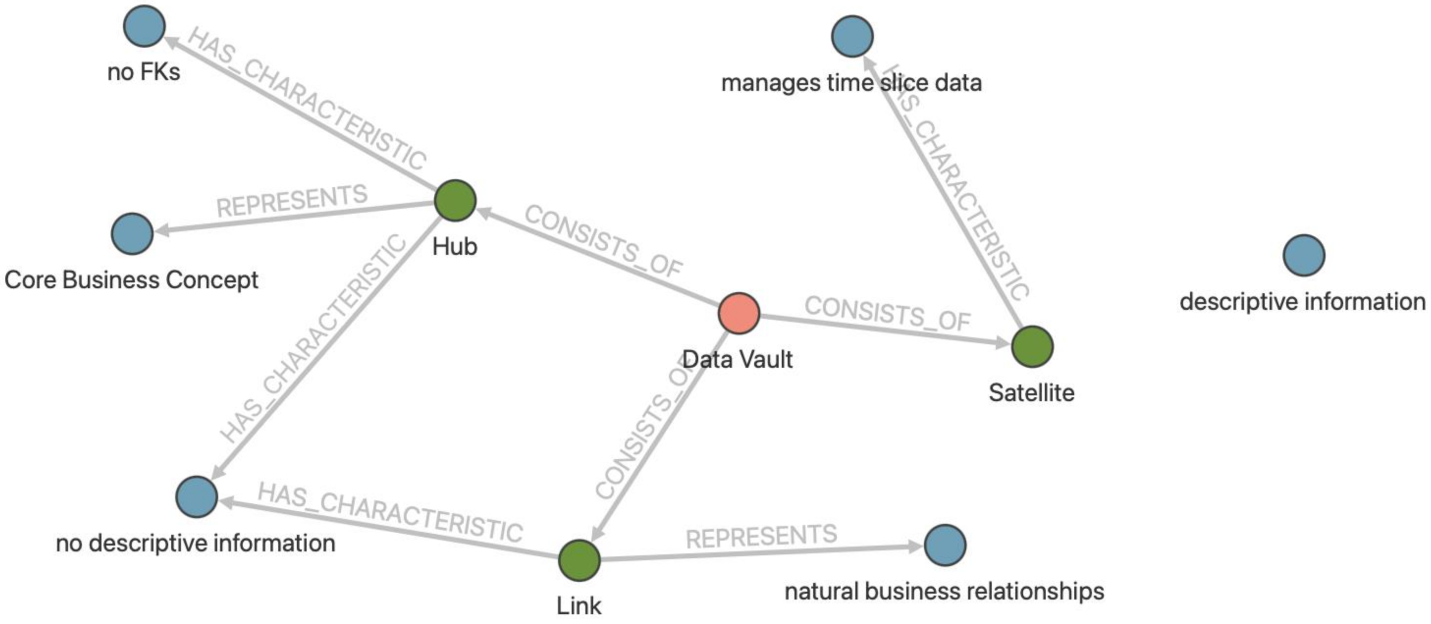
The data vault consists of three core components, the Hub, Link and Satellite. Above all other DV Program rules and factors, the commitment to the consistency and integrity of these constructs is paramount to a successful DV Program.

The Hub represents a Core Business Concept such as Customer, Vendor, Sale or Product. The Hub table is formed around the Business Key of this concept and is established the first time a new instance of that …

A Link represents a natural business relationships between business keys and is established the first time this new unique association is presented to the EDW. It can represent an association between several Hubs and sometimes other Links. It does maintain a 1:1 relationship with the …

The Satellite contains the descriptive information (context) for a business key. There can be several Satellites used to describe a single business key (or association of keys) however a Satellite can only describe one key (Hub or a Link). There is a good amount of flexibility afforded the modelers in how they design and build Satellites. Common approaches include …

Links represent all relationships and the Satellites provide all the context and changes over time.


no FKs
HAS_CHARACTERISTIC
manages time slice data
HAS_CHARACTERISTIC
REPRESENTS
Hub
CONSISTS_OF
Core Business Concept
HAS_CHARACTERISTIC
descriptive information
Data Vault
CONSISTS_OF
Satellite
CONSISTS_OF
HAS_CHARACTERISTIC
no descriptive information
REPRESENTS
natural business relationships
Link

Issues?

- If the document is large -> context window might get too small
- Agent solution
  - Both in transform and in query phase
    - Pages 1-50 Agent1
    - Pages 51-70 Agent2
    - …
    - Combine the retrieved context

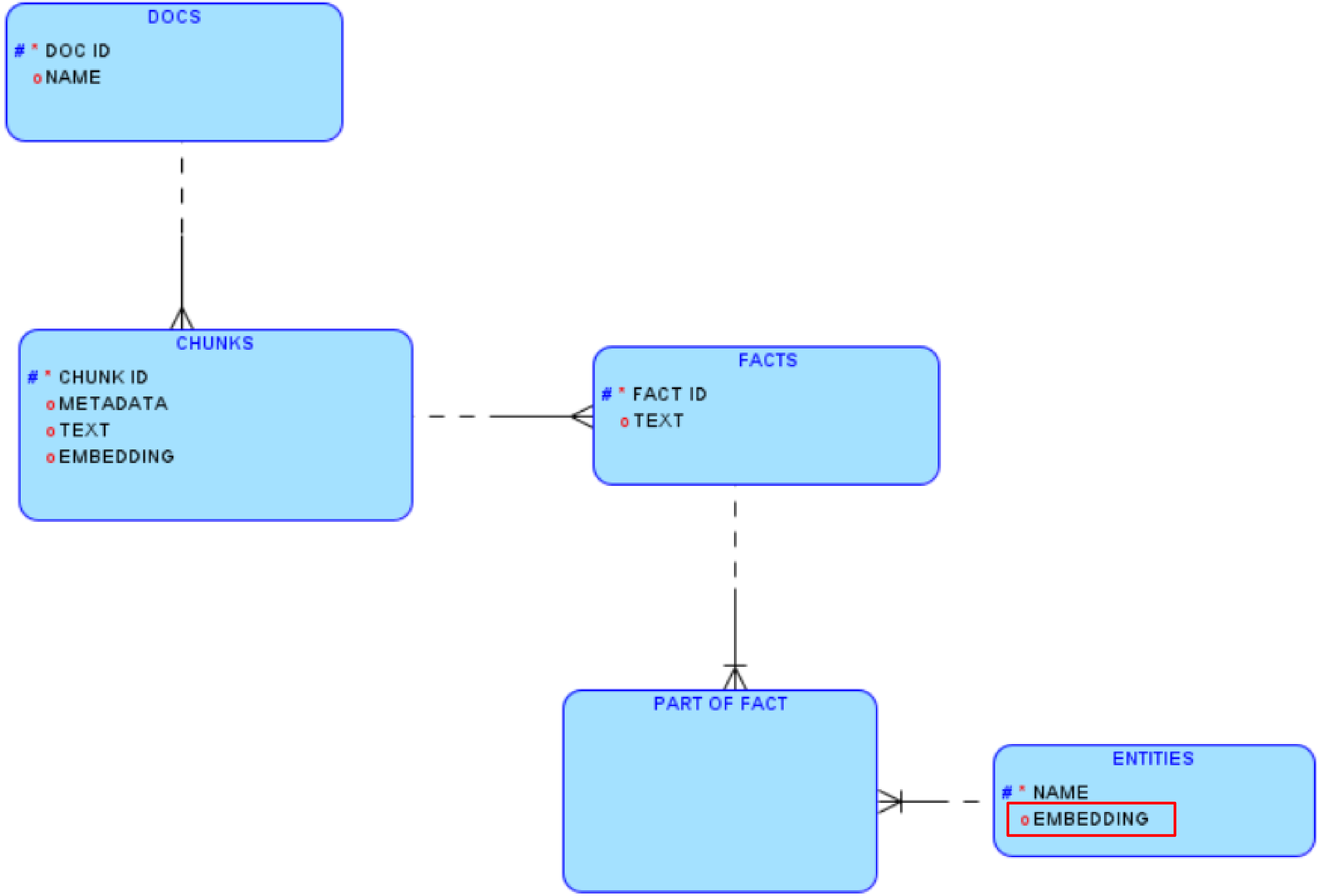
Combine vectors and graph?

- If the graph is very large the RAG solution might not work efficiently
- Combine the techniques!
  - Do a vector or keyword search to find an initial set of nodes.
  - Traverse the graph to bring back information about those related nodes.

  - GraphML

Adding vectors for entities

- Now both entities and chuncks embedded (vectorized)


DOCS
# * DOC ID
o NAME
CHUNKS
# * CHUNK ID
o METADATA
o TEXT
o EMBEDDING
FACTS
# * FACT ID
o TEXT
PART OF FACT
ENTITIES
# * NAME
o EMBEDDING

Demo

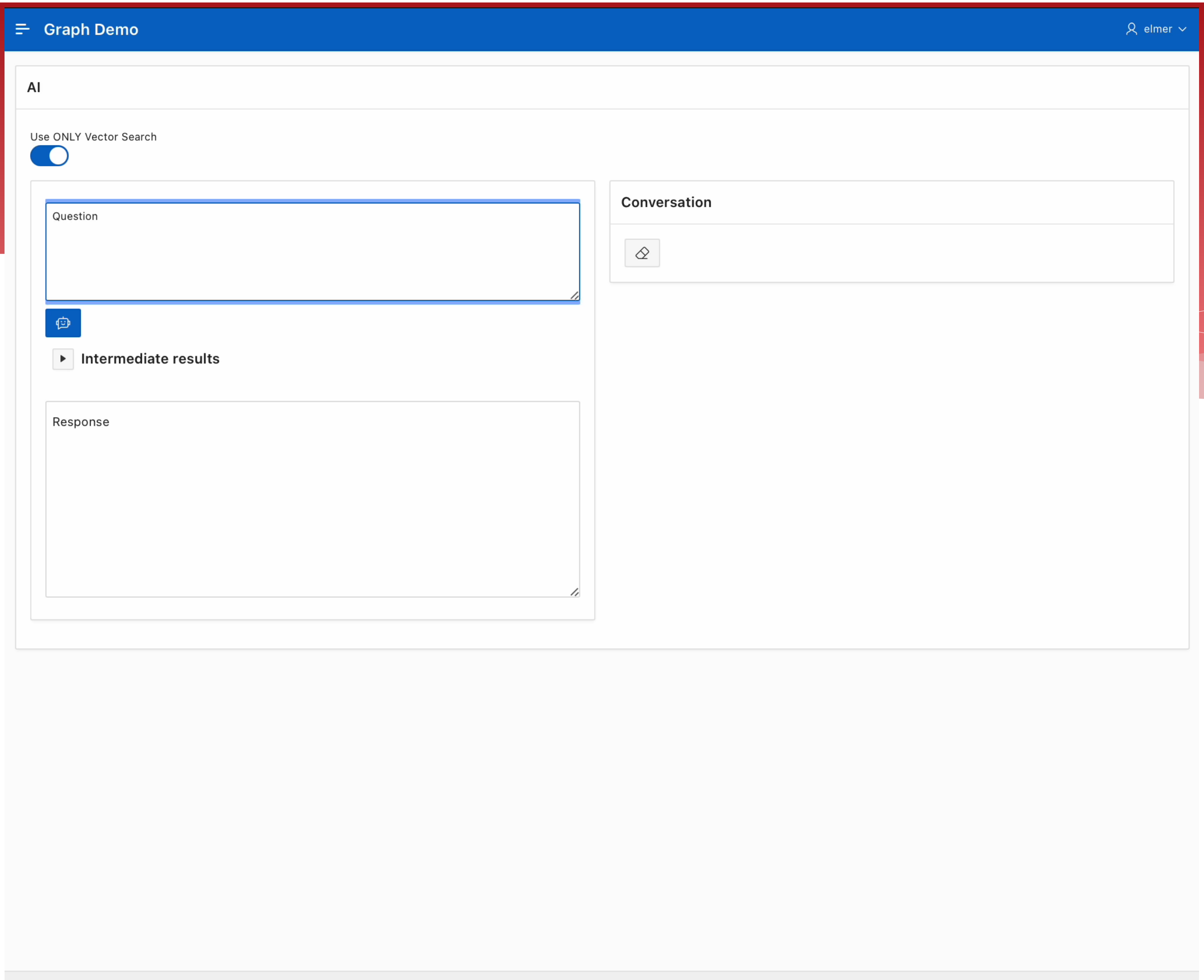
- For this we used APEX and PL/SQL

The context

- PDFs uploaded:
- https://docs.oracle.com/en/database/oracle/machine-learning/oml4sql/21/dmcon/oracle-machine-learning-sql-concepts.pdf
- https://docs.oracle.com/en/database/oracle/machine-learning/oml4py/1/mlpug/oracle-machine-learning-python-users-guide.pdf
- About 600 pages

Using only vectors

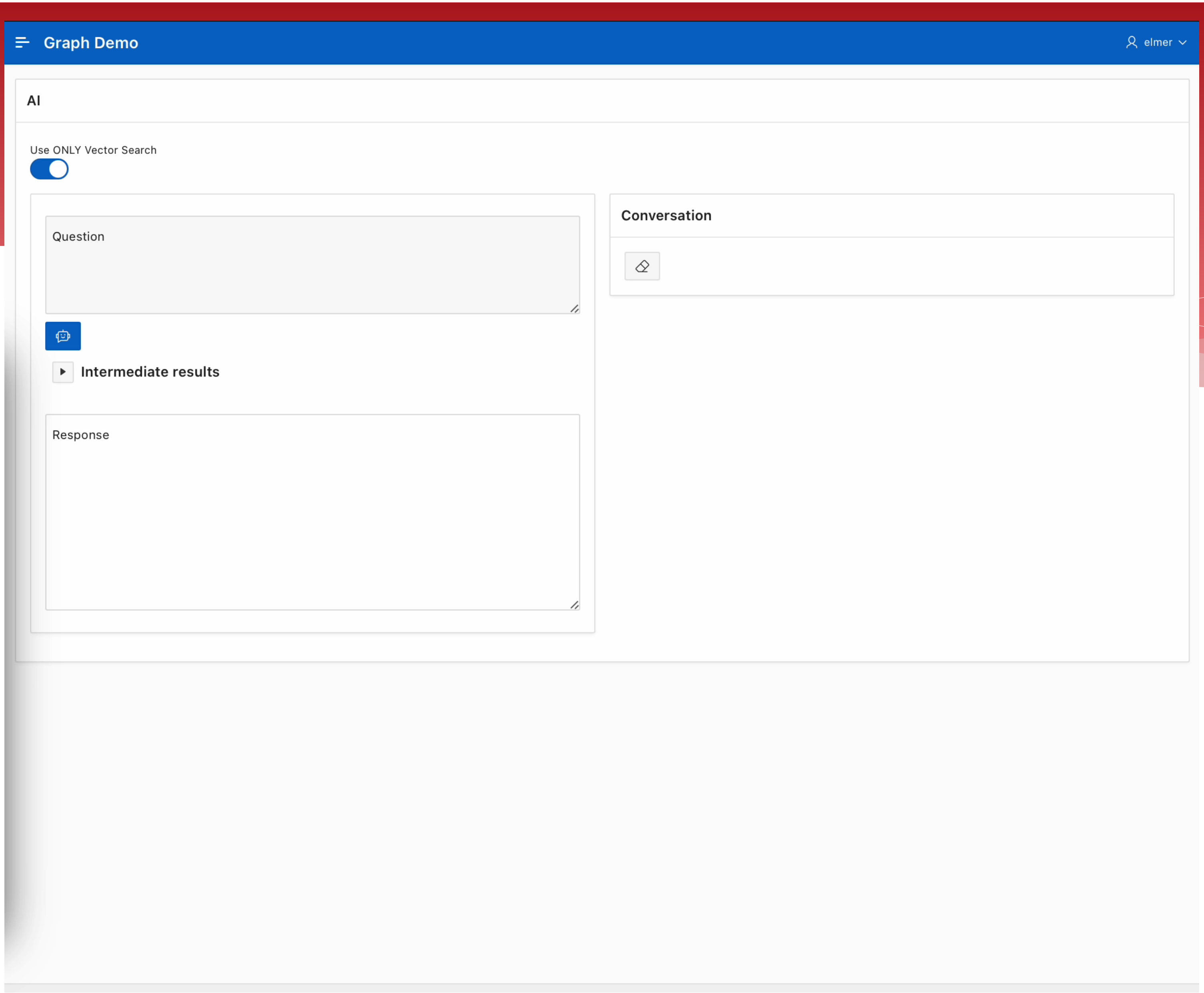
- Demo



Copyright © Miracle Finland Oy

Using both Vectors and Graph

- Demo



Graph Demo
elmer
AI
Use ONLY Vector Search
Question
Intermediate results
Response
Conversation

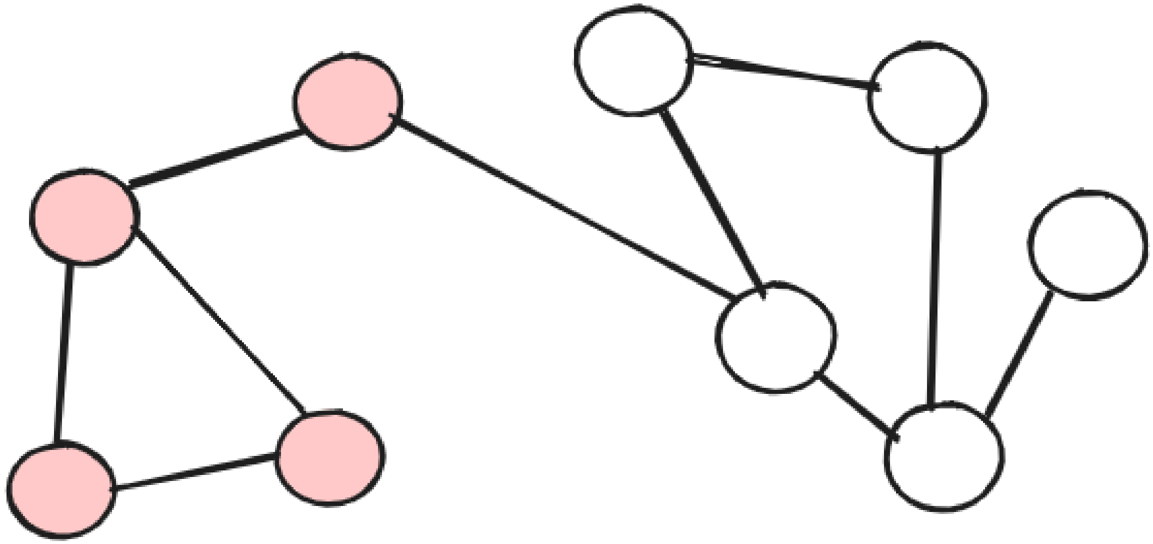
Other techniques for GraphRAG?

- Plenty of them
- And improving fast
- For example embedding nodes or entire graphs, and then using a similarity search

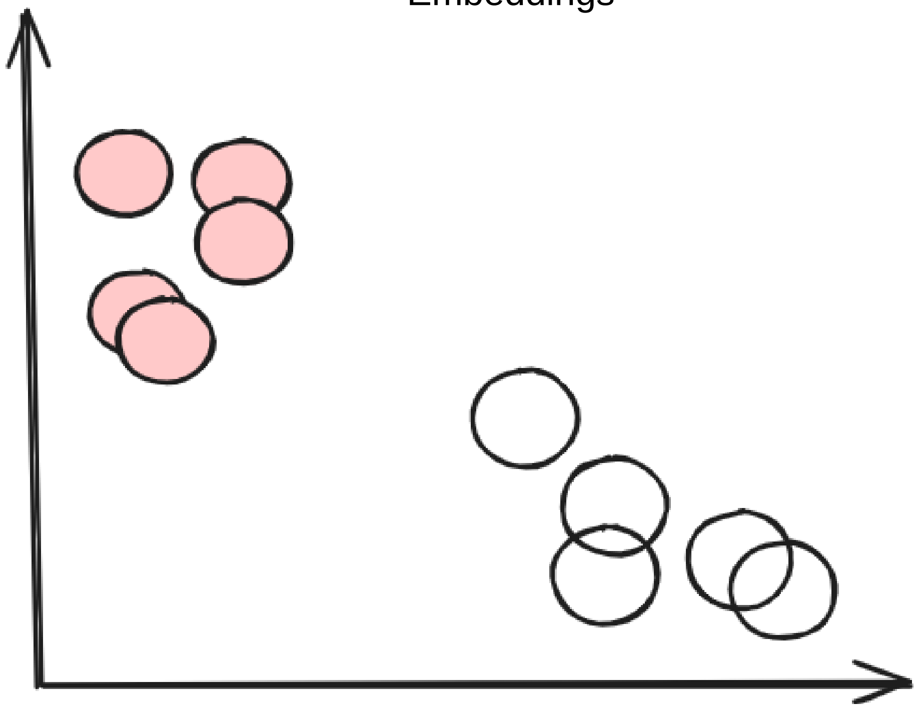
MIRACLE
Miracle Finland Oy

Node embedding

Graph



Embeddings



Embedding graphs,
Node embedding techniques

- Individual nodes as vectors
- **But** also including the context in the graph
- These techniques aim to **capture their essence and relationships** of a node through numerical encoding, a vector.

Examples of node embedding techniques

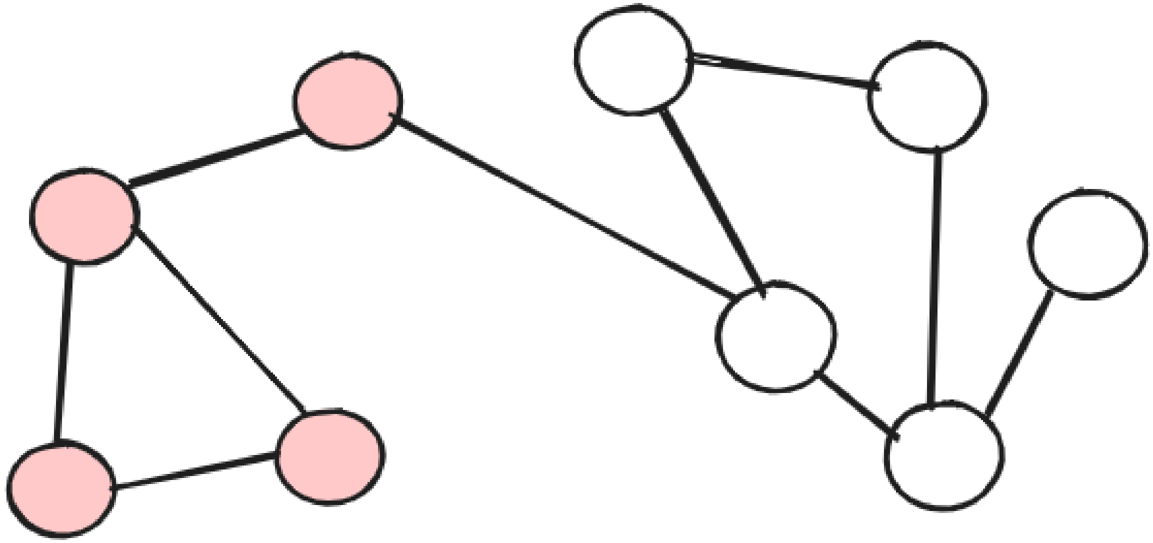
- Factorization based
  - the relationships between nodes as a matrix
  - factorize this matrix to obtain the embedding
- Random Walk based
- Deep Learning based

Embedding graphs

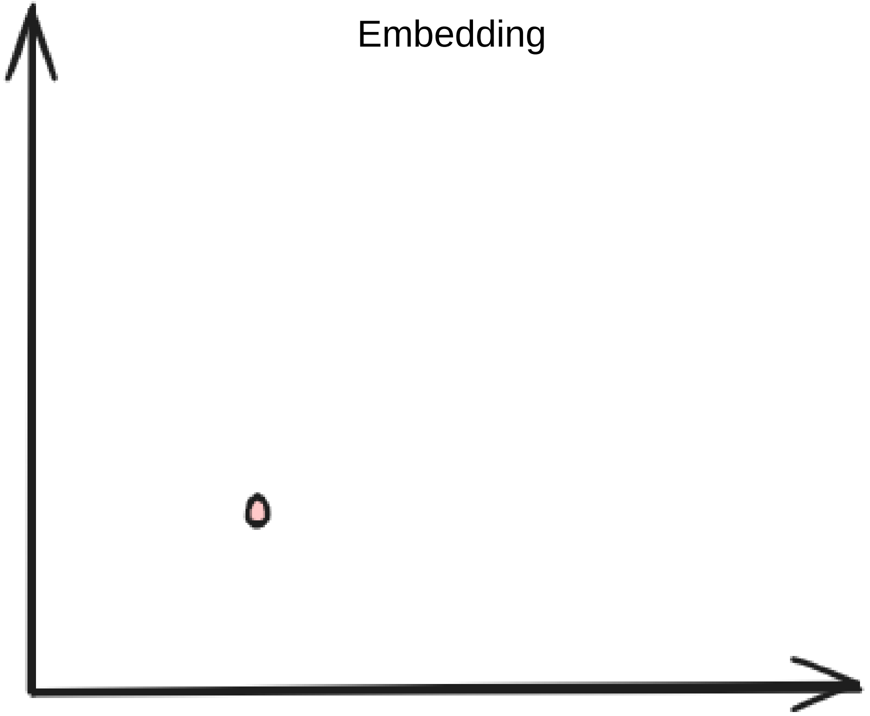
- Graph algorithms capture general **structure** of the **entire network** and the **interactions** in it.

Embedding graphs

Graph



Embedding



Examples of Graph embedding algorithms

- Graph Convolutional Networks (GCNs)
- Graph Attention Networks (GATs)
- Graph Neural Networks (GNNs)

Conclusions

- Graphs are powerful
- Provides grounding for search
- Evolving topic

Thank you!

QUESTIONS?

Hans.viehmann@oracle.com

Heli.helskyaho@miracleoy.fi

Linkedin.com/in/helihelskyaho/

Twitter: @HeliFromFinland

Blog: Helifromfinland.com

MiRACLE

Miracle Finland Oy