

# VectorDB & RAG

VLDB Lab.

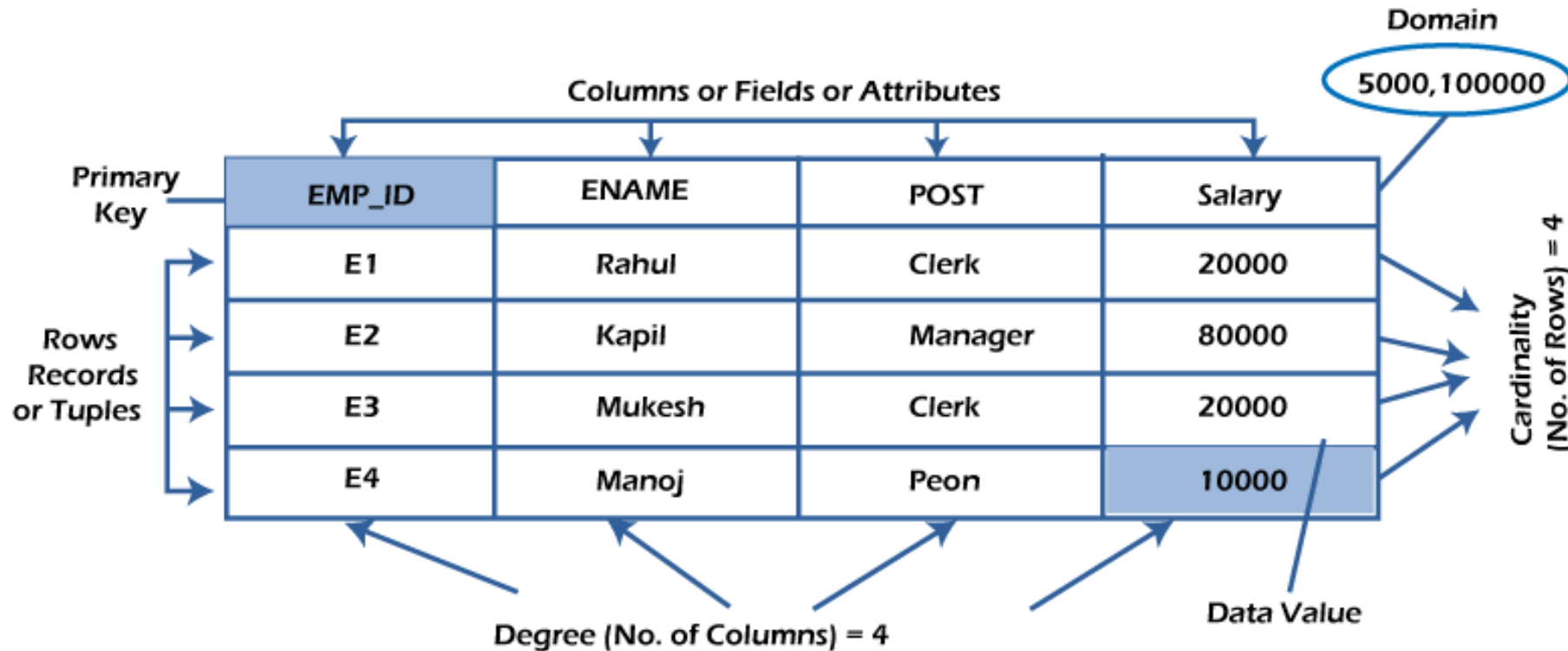
Professor Sangwon Lee

# Contents

- RDBMS v.s. Vector Database
- RAG란?
- ChromaDB란?
- ChromaDB Getting Started
- RAG System 실습

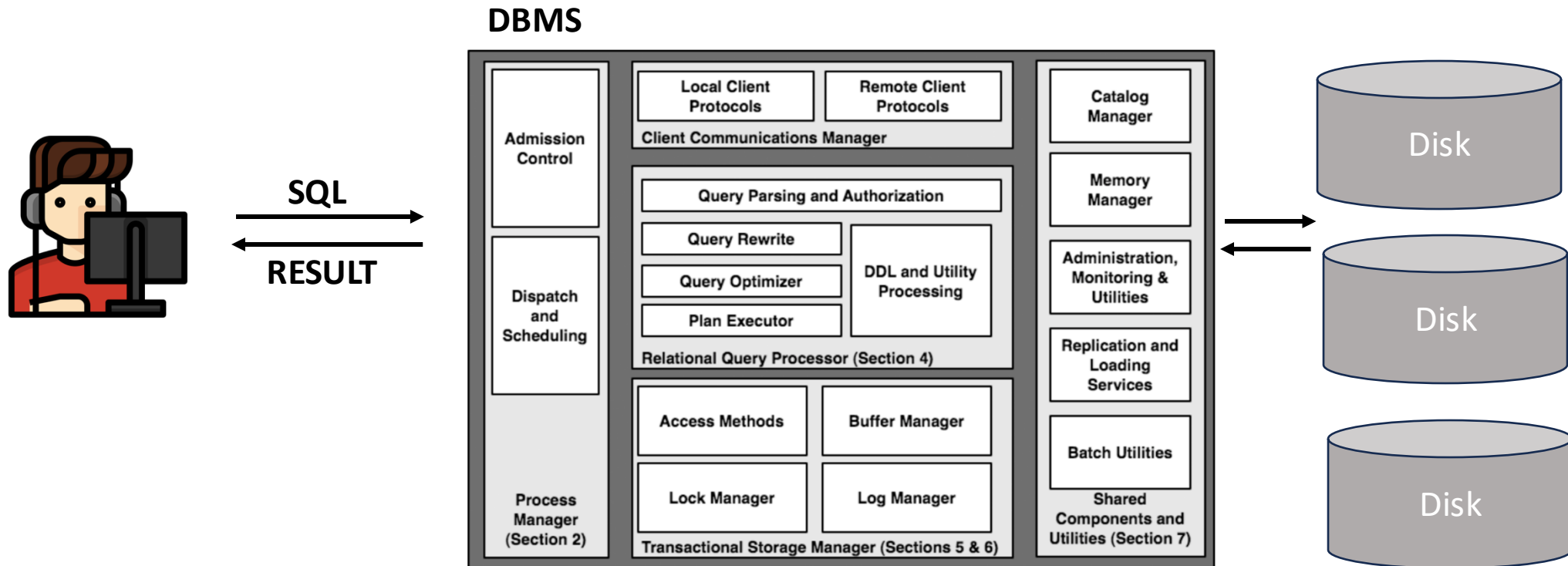
# Relational Database Management System (RDBMS)

- 테이블 형태로 정보를 저장하는 데이터베이스



# Relational Database Management System (RDBMS)

- 테이블 형태로 정보를 저장하는 데이터베이스



# Other Database Systems

- NoSQL



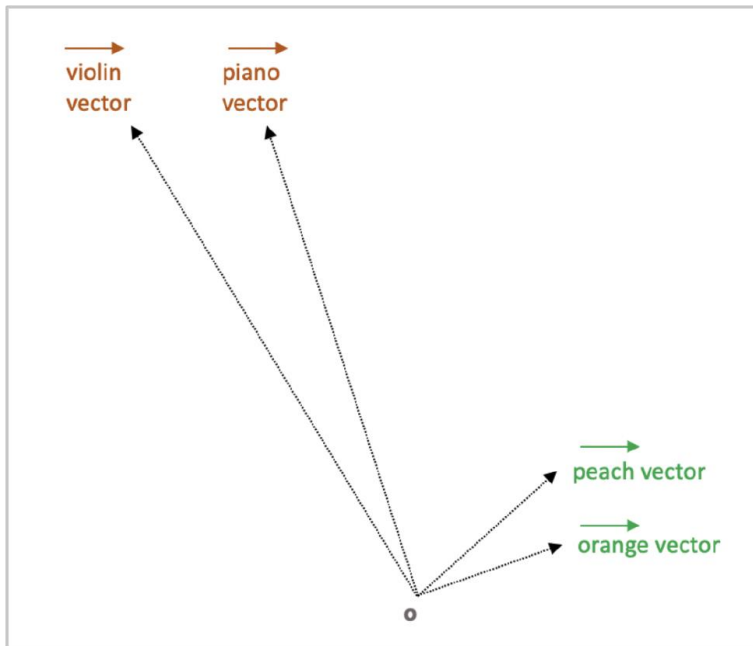
- Cloud Data Warehouse Solution



- Etc.

# Vector Database: When and Why

- **벡터란?** 데이터 객체의 수치 표현
- **벡터 임베딩:** 단어와 문장, 기타 데이터를 의미와 관계를 포착하는 숫자로 변환하는 방법



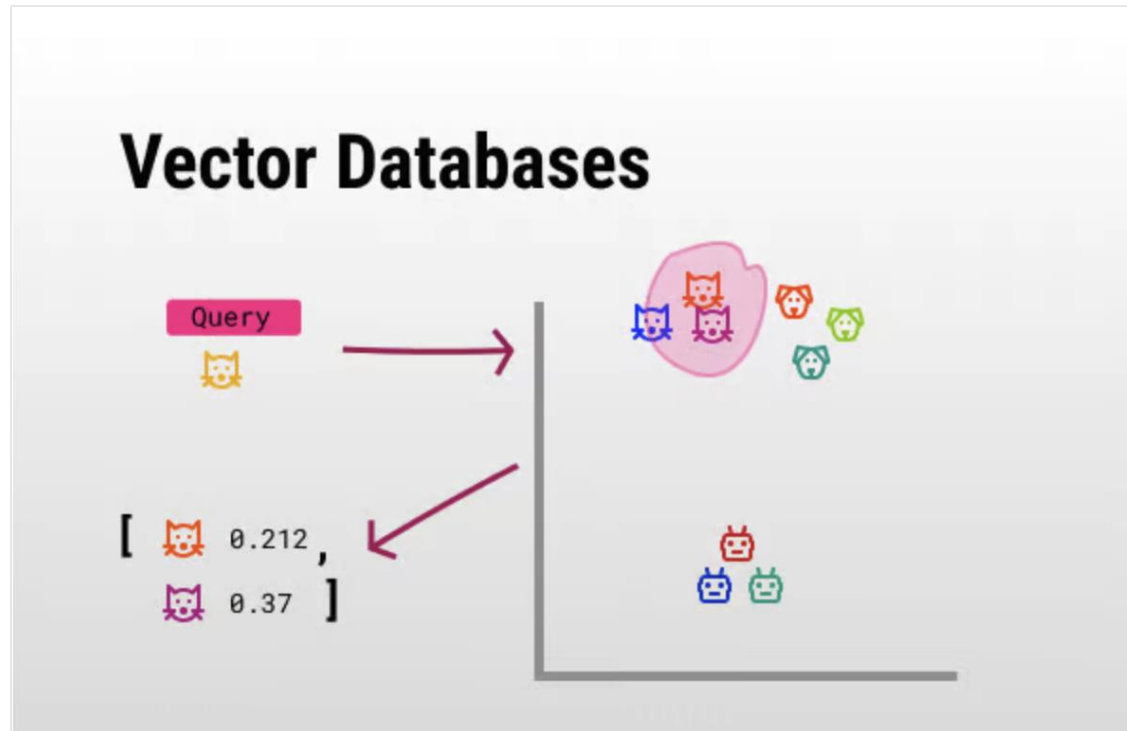
나는	[1, 0, 0, 0, 0]
임베딩	[0, 1, 0, 0, 0]
공부를	[0, 0, 1, 0, 0]
하고	[0, 0, 0, 1, 0]
있다	[0, 0, 0, 0, 1]

나는	[0.123, -0.411, 0.345, 0.198, 0.757]
임베딩	[0.841, 0.742, 0.956, 0.312, 0.149]
공부를	[2.314, 0.742, 1.233, 0.534, -0.015]
하고	[-1.434, 0.742, 0.231, 0.864, 0.508]
있다	[0.389, 0.742, 1.412, 1.218, -1.113]

# Vector Database: When and Why

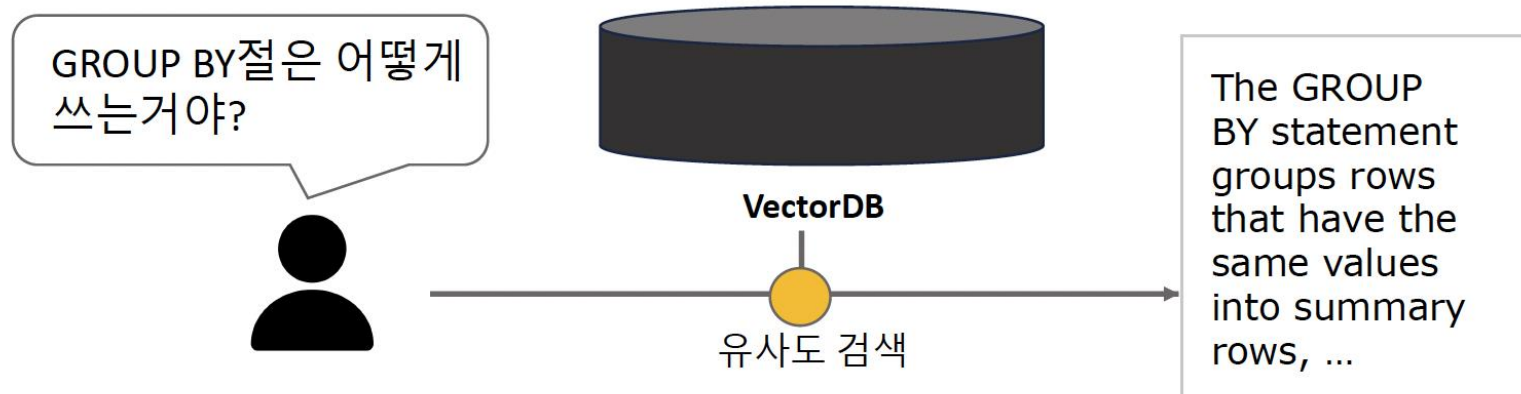
- 벡터 데이터베이스

- 정보를 벡터로 저장하는 데이터베이스.
- 이러한 벡터를 기반으로 유사성 검색(Similarity Search)과 같은 작업을 효율적으로 수행할 수 있도록 설계된 데이터베이스



# Vector Database: When and Why

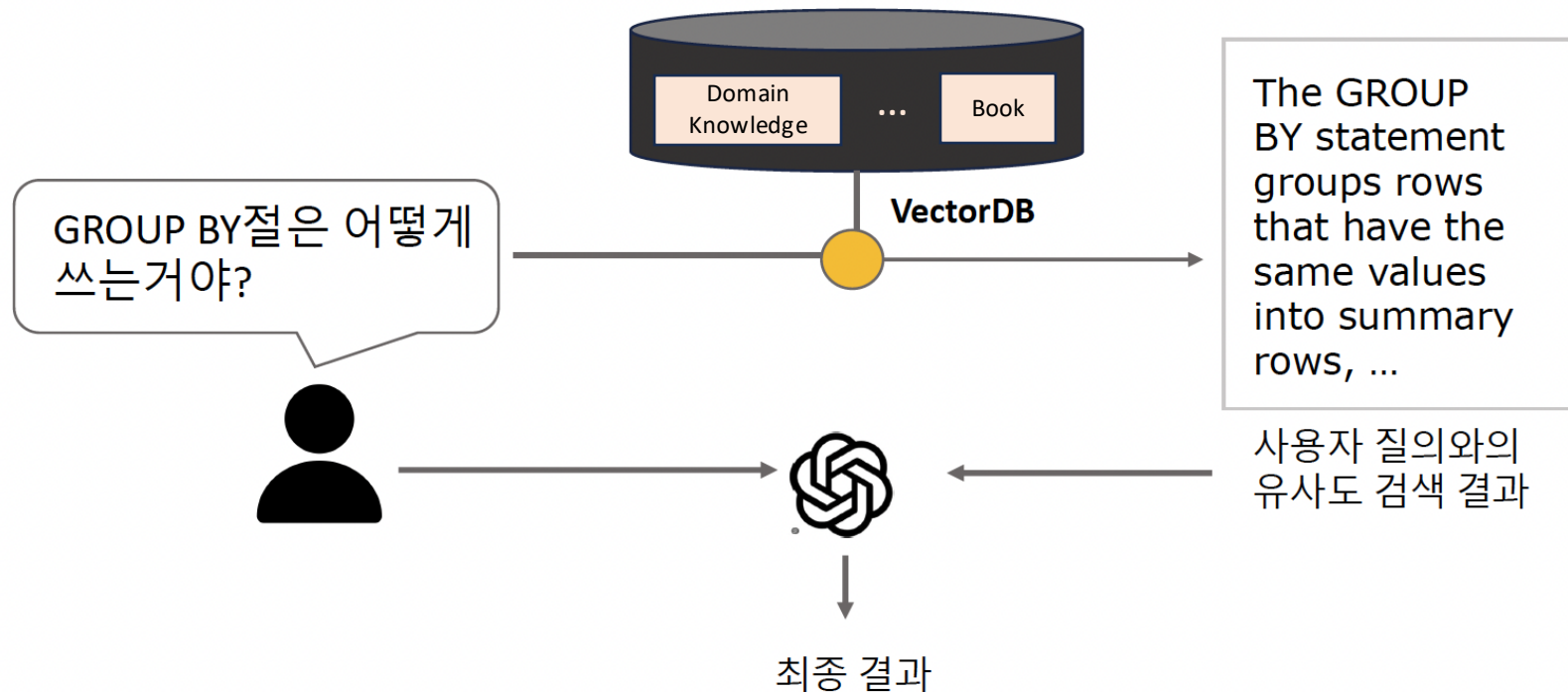
- 벡터 데이터베이스에서 쿼리는 무엇이고, 어떤 데이터를 반환하는 걸까?
  1. 벡터 쿼리 생성: 사용자가 검색하려는 내용이나 질문의 벡터 형태의 쿼리 생성
  2. 벡터 유사성 검색: 벡터 데이터베이스에서는 쿼리 벡터와 저장된 벡터 간의 유사성 계산(코사인 유사도, 유클리드 거리 계산 등)
  3. 쿼리 수행: 사용자가 쿼리를 제출하면, 벡터 데이터베이스는 저장된 벡터와 쿼리 벡터 간의 유사성을 계산



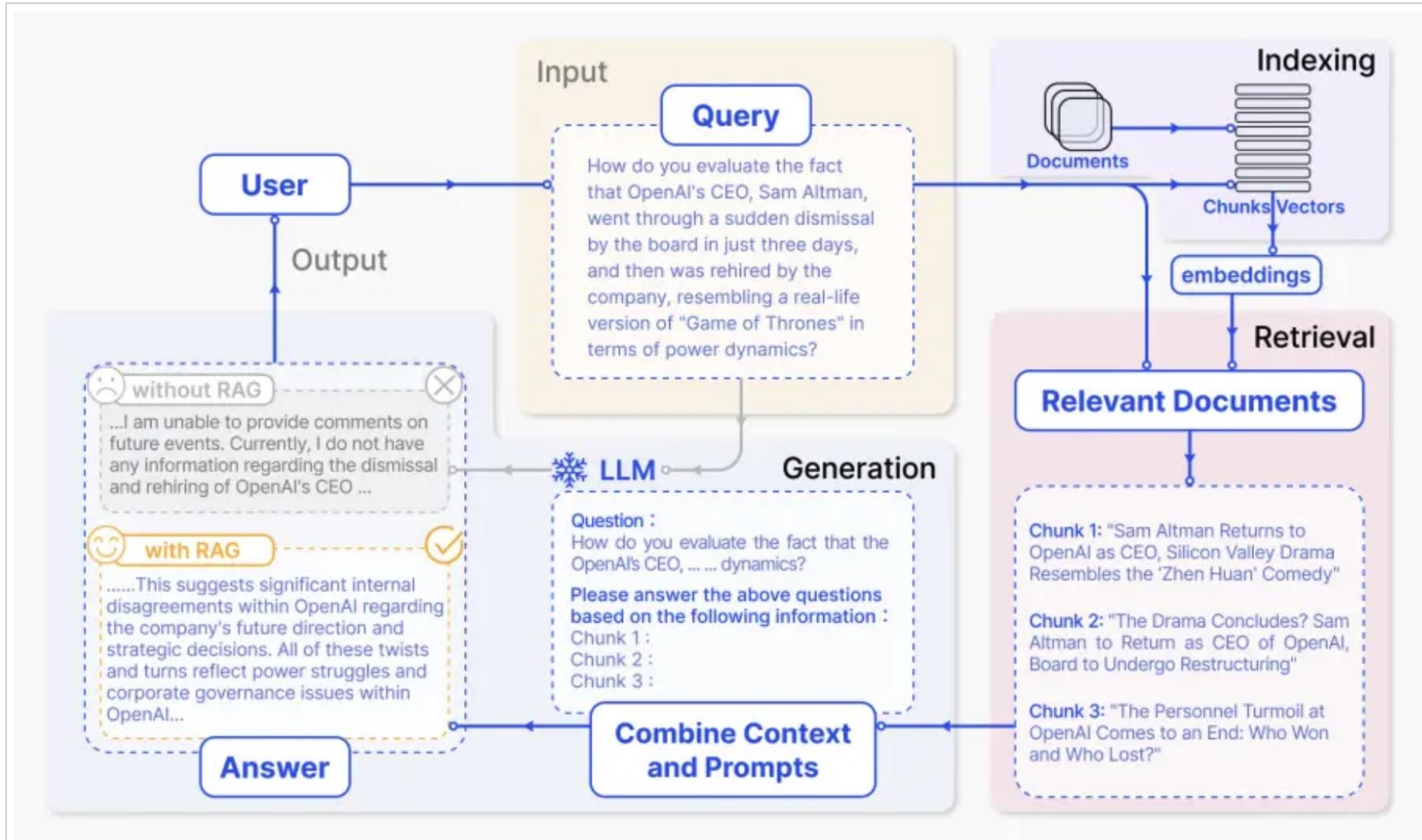


# Retrieval-Augmented Generation (RAG)

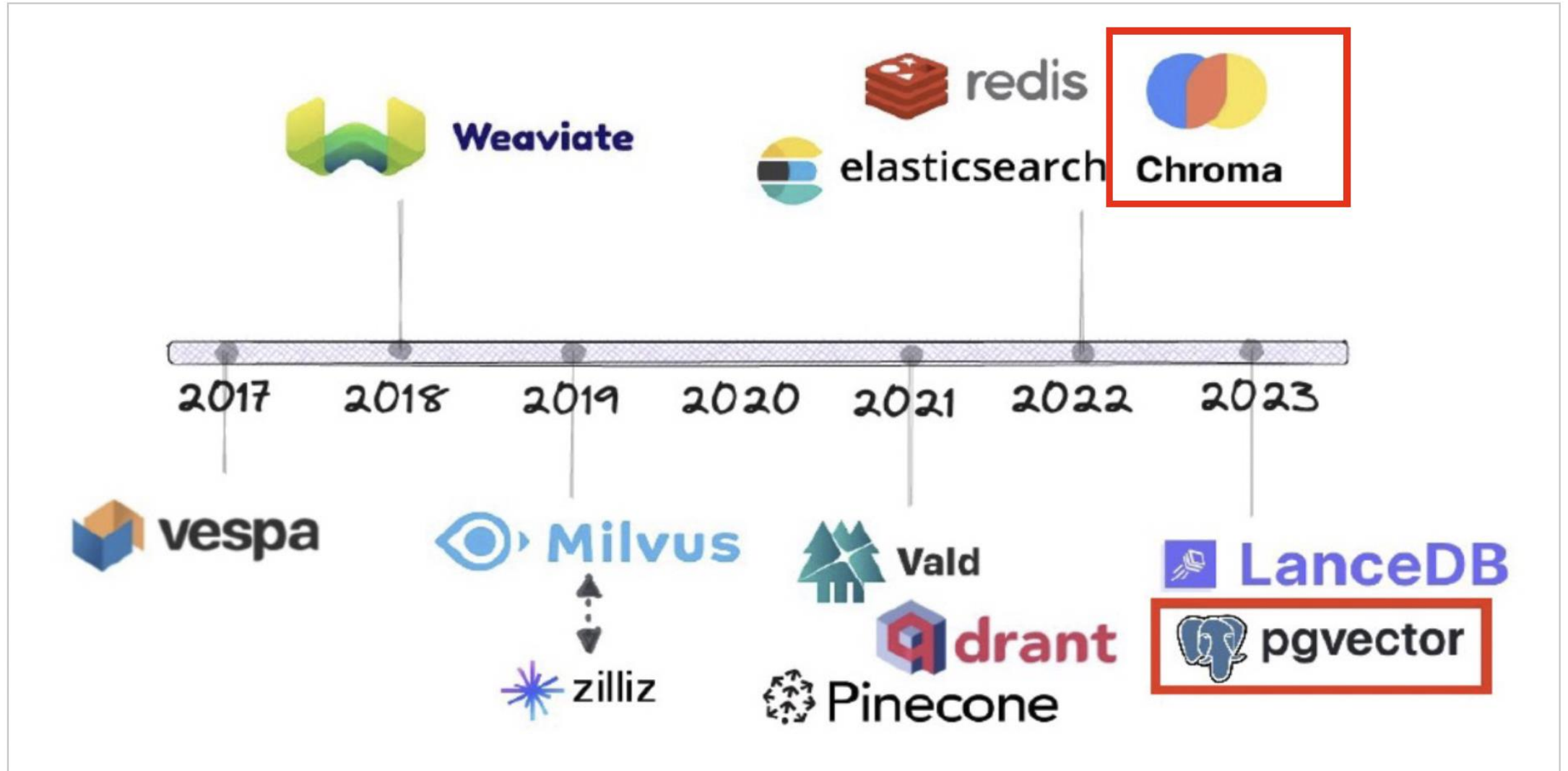
- 생성형 모델과 검색 기반 모델을 결합한 자연어 처리 접근 방식
- 정보 검색과 텍스트 생성을 통합하여 보다 풍부하고 정확한 답변을 생성하는데 중점을 둠.



# Retrieval-Augmented Generation (RAG)

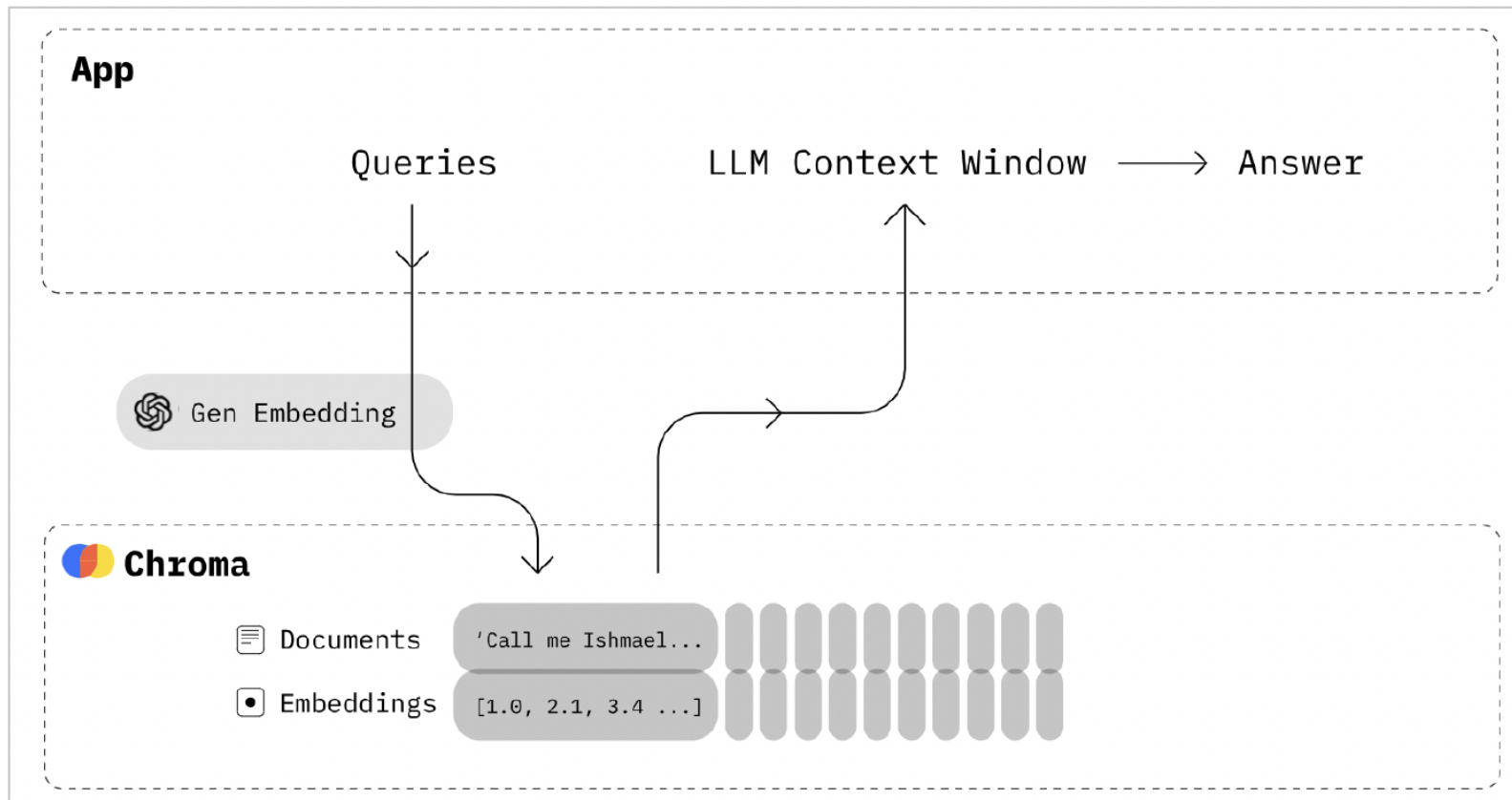


# Timeline



# ChromaDB


- AI-Native Open Source Vector Database
- 플러그형으로 만들어 LLM application을 쉽게 구축할 수 있도록 함.



# ChromaDB Getting Started

<https://docs.trychroma.com/docs/overview/getting-started>


## Install

 Command Line

 Copy Code

```
1 pip install chromadb
```


## Create a Chroma Client

 python

 Copy Code

```
1 import chromadb
2 chroma_client = chromadb.Client()
```

## Create a Collection \*\*Collection: 벡터 데이터를 저장하는 기본 단위

 python

 Copy Code

```
1 collection = chroma_client.create_collection(name="my_collection")
```

# ChromaDB Getting Started

<https://docs.trychroma.com/docs/overview/getting-started>

## Add some text documents to the collection

<> python

Copy Code

```
1 collection.add(  
2     documents=[  
3         "This is a document about pineapple",  
4         "This is a document about oranges"  
5     ],  
6     ids=["id1", "id2"]  
7 )
```

## Query the collection

<> python

Copy Code

```
1 results = collection.query(  
2     query_texts=["This is a query document about hawaii"], # Chroma will embed this  
3     n_results=2 # how many results to return  
4 )  
5 print(results)
```



# RAG Practice

- ChromaDB 기초 실습 [https://github.com/snu-vldb-ta/SNU-BigData-Fintech-2025-1H/blob/main/5/chromadb\\_getting\\_started.ipynb](https://github.com/snu-vldb-ta/SNU-BigData-Fintech-2025-1H/blob/main/5/chromadb_getting_started.ipynb)
- ChromaDB RAG 실습 [https://github.com/snu-vldb-ta/SNU-BigData-Fintech-2025-1H/blob/main/5/chromadb\\_RAG.ipynb](https://github.com/snu-vldb-ta/SNU-BigData-Fintech-2025-1H/blob/main/5/chromadb_RAG.ipynb)