ORACLE

N.EX.T 정기 기술 세미나
생성형 AI 시대에 오라클 3개 전략

김태완

Cloud Engineer Team

Oracle Korea

2024.11.16

Oracle의 전략-1

Oracle은 자체 LLM을 개발하지 않습니다.

Oracle Generative AI 전략 & 차별성

Model

DATA

Oracle Database 23C

ADW ATP EXACI EXACC

OpenSearch

Heatwave

INFRASTRUCTURE

H100/A100 /A10 GPU

Cloud Native OKE

Super Cluster

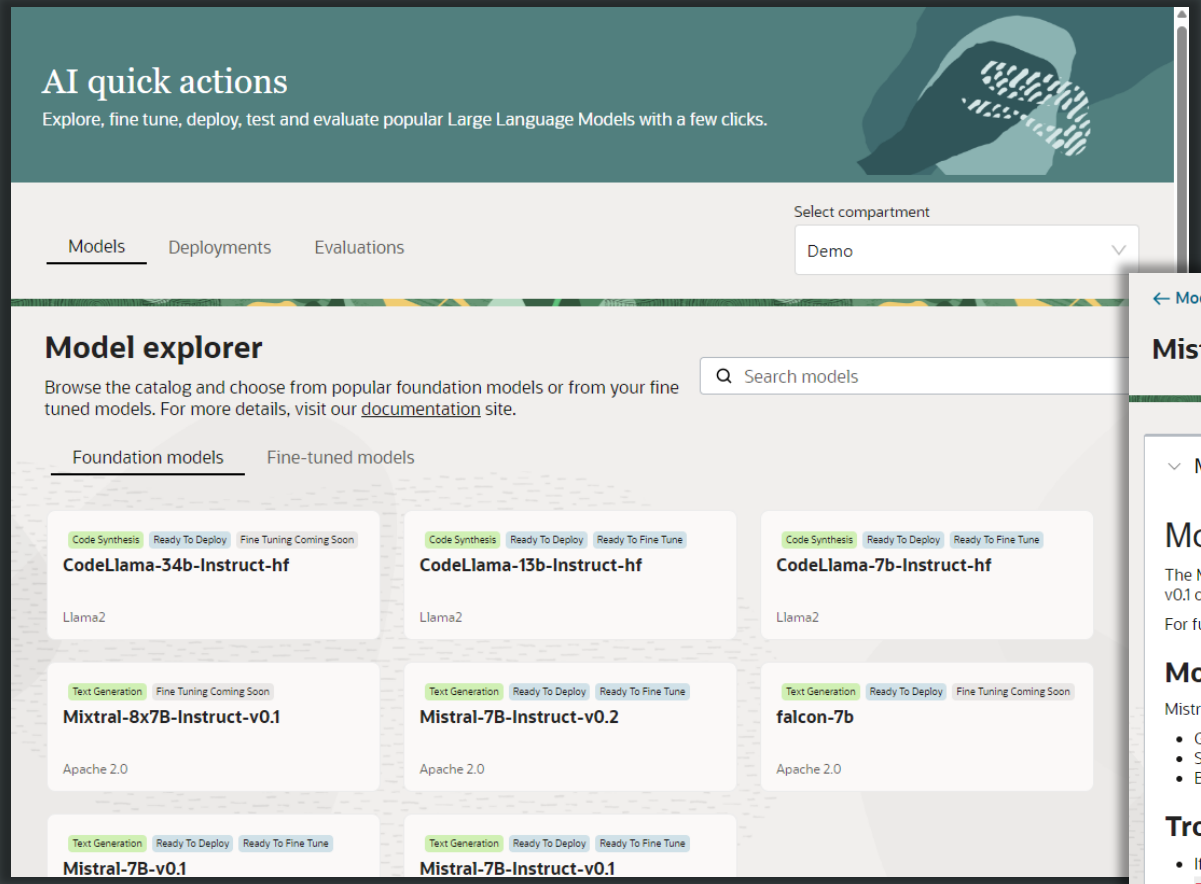
Chatbot

Oracle Enterprise LLM/Generative AI Platform

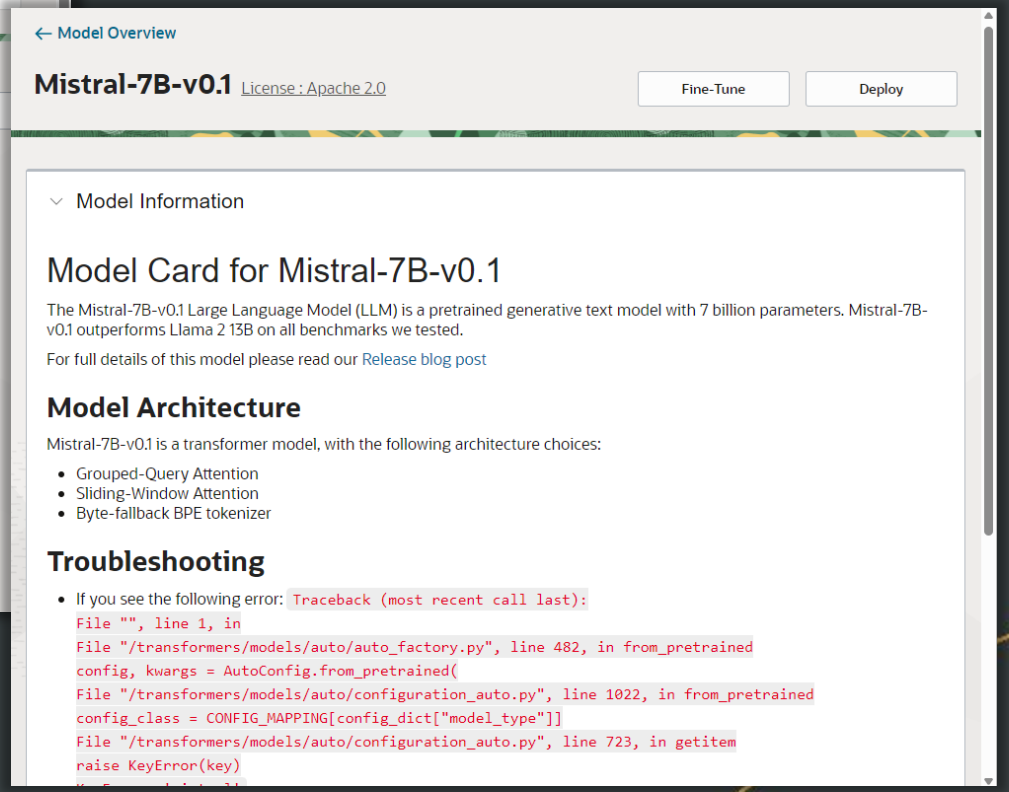
OCI Supercluster

- AI 모델을 학습시키기 위해서는 대규모 고성능 GPU클러스터 필요
- 최신 OCI Supercluster는 13만개 이상의 GPU가 함께 작동하고 수백명의 고객에게 최고 성능 GPU클러스터 제공
- OCI Supercluster는 액체 냉각, RDMA 네트워크, 고속 파일 스토리지를 결합하여 가장 까다로운 AI 워크로드를 지원
  - 초당 300 쿼드릴리온 (310,000조, 310경) 패킷 처리 능력
  - 104 페타비트/초의 Aggregate bandwidth
  - 기존 공랭식과 AI용 액체 냉각 복합 적용

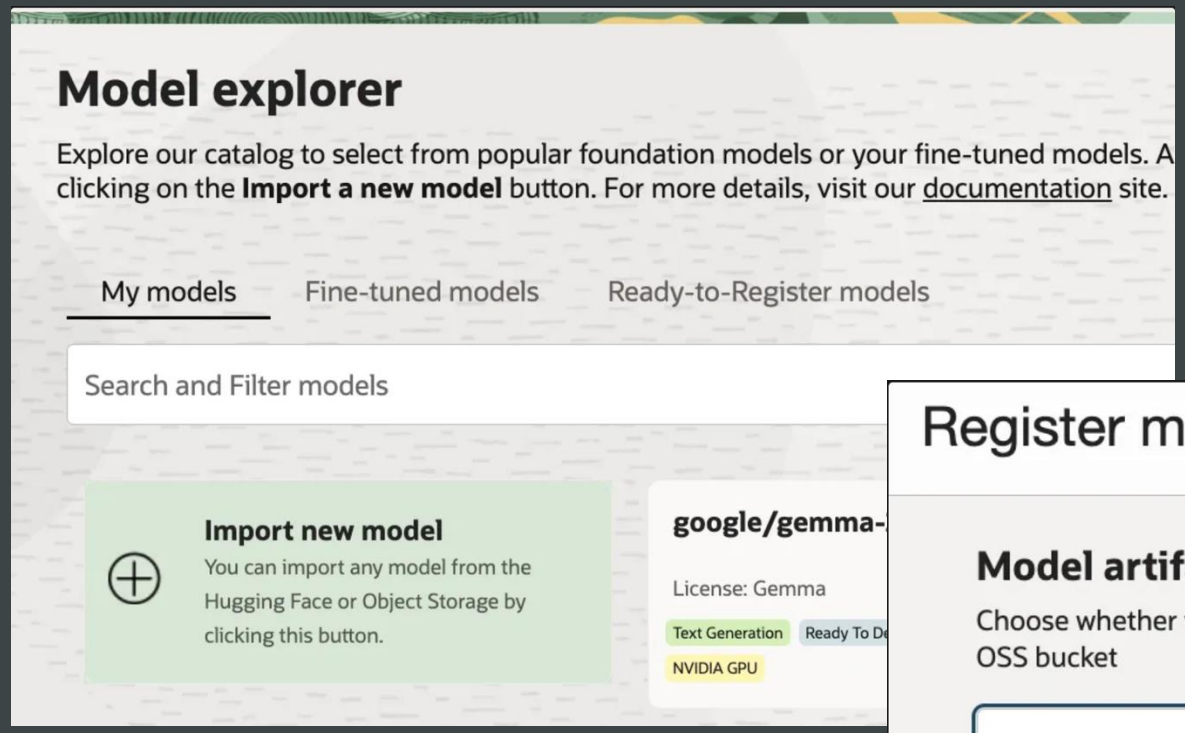
Hugging Face 통합: Data Science AI Quick Action



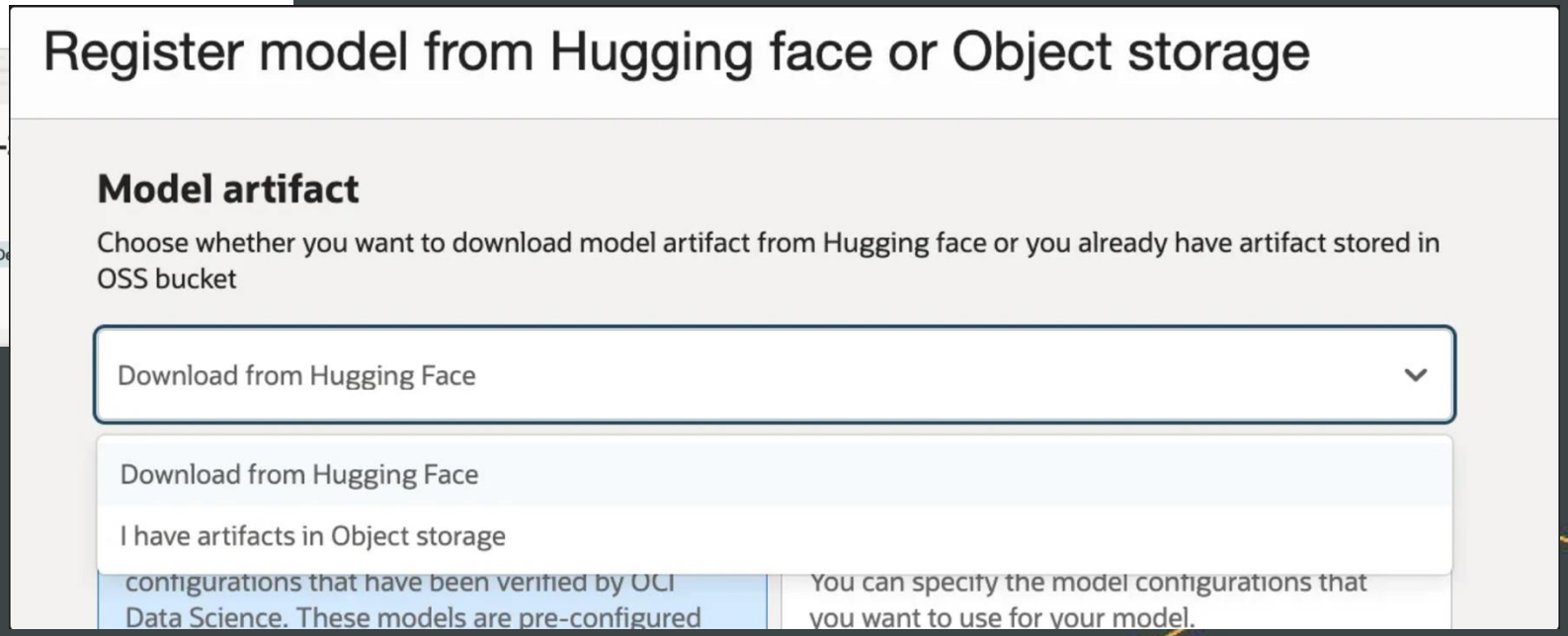
Hugging Face 오픈 모델의
NoCode 기반 통합 및 배포



Hugging Face 통합 :Data Science AI Quick Action



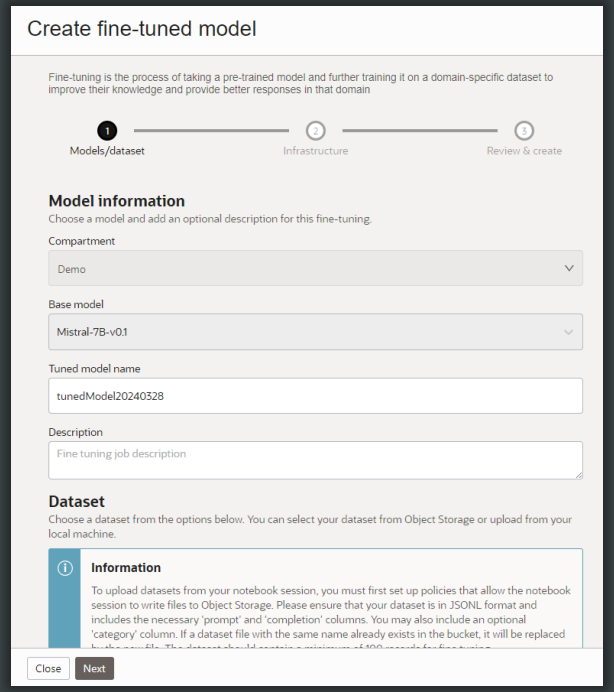
No-code 솔루션: 오픈소스 모델 배포, 미세 조정, LLM 평가



Hugging Face 통합 :Data Science AI Quick Action

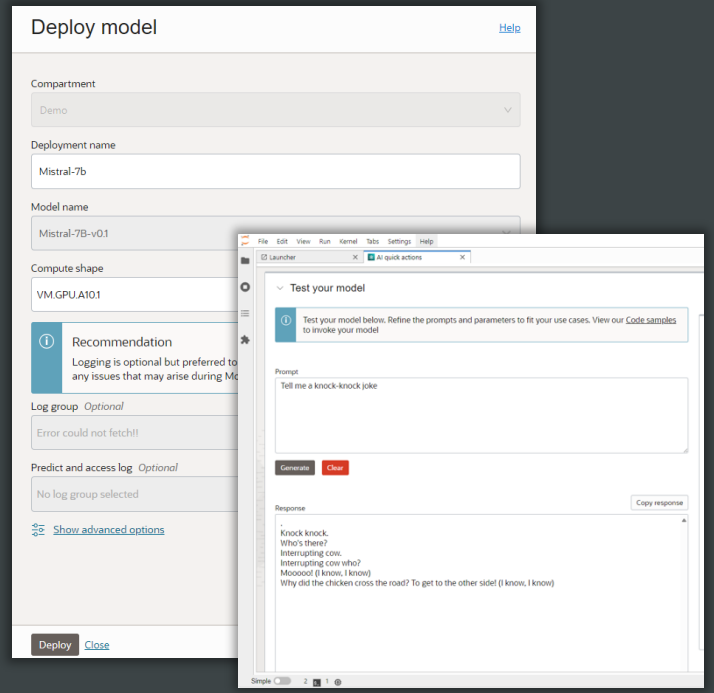
미세 조정 UI

사용자 데이터로 미세 조정 프로세스를 통해 모델 특화, 미세 조정된 모델은 사용자의 모델 저장소에 저장



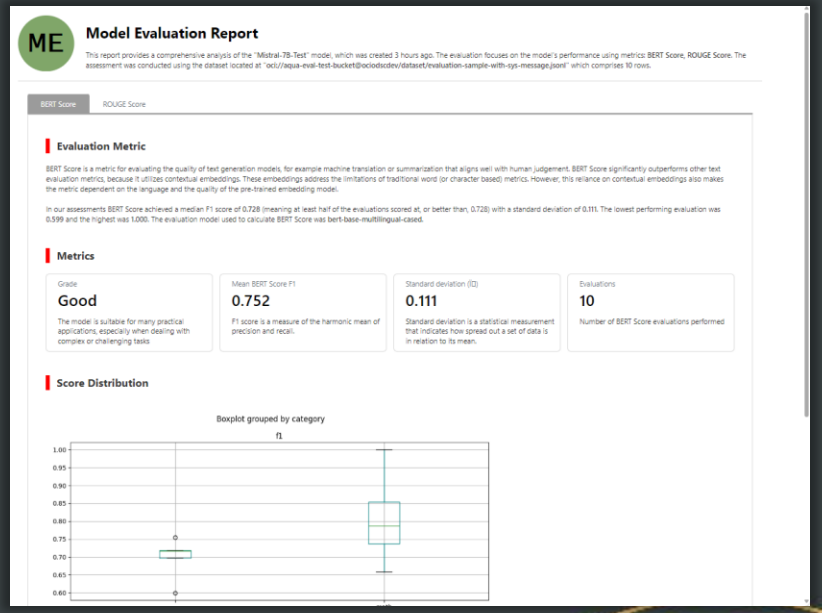
모델 배포 및 테스트

실시간 추론 엔드포인트 지원, 모델 상호작용 검증용 테스트 환경 제공

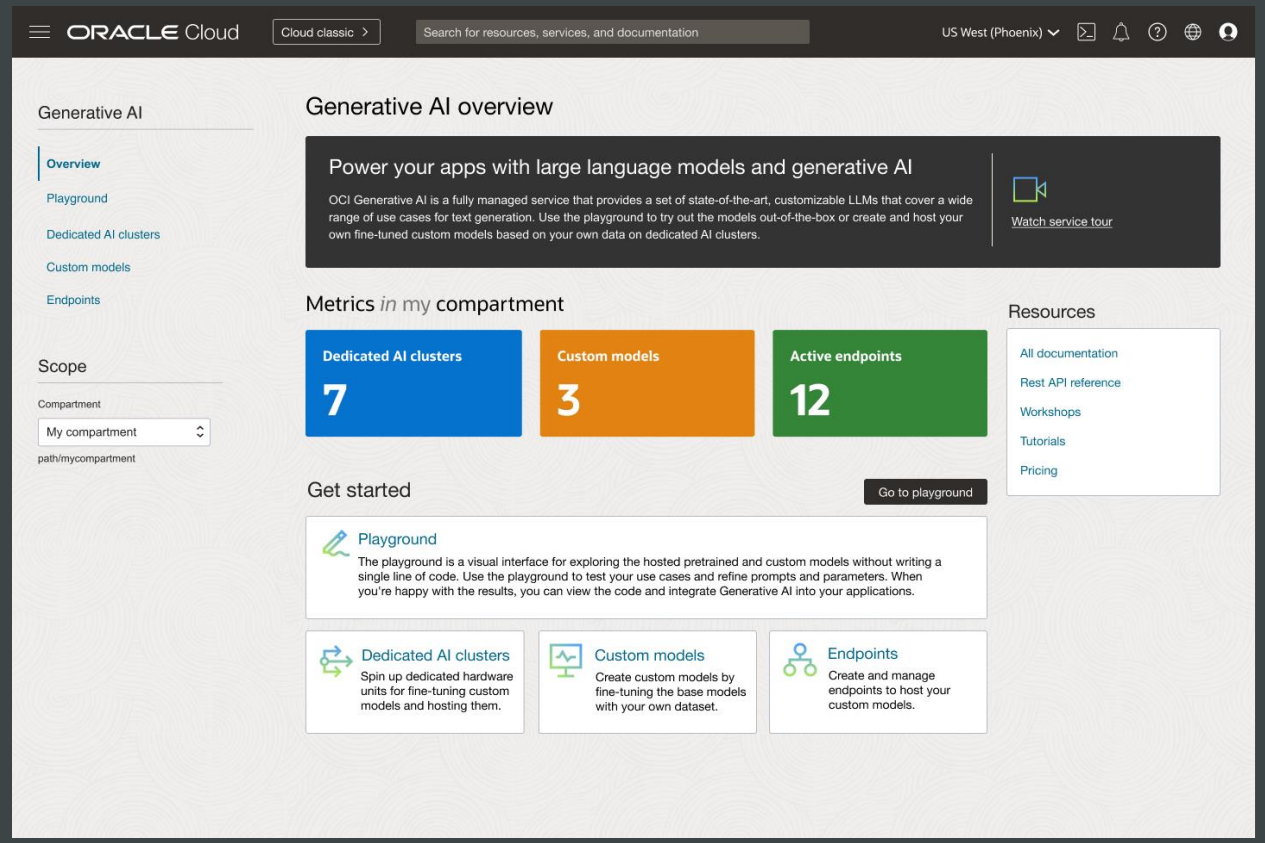


모델 평가

BERTScore, ROUGE 및 기타 지표를 사용하여 자세한 성능 보고서로 모델을 비교



완전 관리형 LLM 서비스: Generative AI



cohere

+

OCI Generative AI 지원 모델

New **Llama-3 70B**
Llama-3.1 405B

Llama-3 **70B** 및 Llama-3.1 **405B** 파라미터 텍스트 생성 모델은 Meta에서 개발한 것으로, 선도적인 오픈 소스 대형 언어 모델(LLM) 연구 및 상업적 용도로 무료로 사용 가능

New **Llama-3.2 90B**
Llama-3.2 11B

Llama-3.2 **90B** & Llam-3.2 **11B** 파라미터 비전 + 텍스트 생성 모델. 멀티 모달 모델, 모델 선택의 다양성을 위한 90B와 11B 모델 지원, Llama-3.2 **90B** 모델은 온-디멘드 및 전용 모델, 파인튜닝 지원

cohere

New **Command R+**

Command R+ is an instruction-following conversational model that performs language tasks at a higher quality, more reliably, andwith a longer context length (up to **128k** tokens) compared to previous models. It is best suited for complex RAG workflows and multi-step tool use. It also has better support than previous model generations for **10 key languages**.

New **Command R**

Command R targets the “scalable” category of models that balance high performance with strong accuracy. It is great for simpler retrieval augmented generation (RAG) and single-step tool use tasks, as well as applications where price is a major consideration. A 16k context length is supported as well as 10 key languages.

Embed

The English and multi-lingual Embedding model (V3) that converts text to vector embeddings. A ‘light’ version of the model exists that is smaller and faster but is slightly less performant (English only).

Oracle의 전략-2

왜 Cohere와 Meta인가?

Cohere: Controllable Model for Enterprise

오라클은 LLM 선도 기업인 Cohere와 긴밀한 파트너쉽을 통해 엔터프라이즈를 위한 Generative AI 서비스 개발

스텐포드에서 수행한 자연어 LLM 모델의 성능을 측정하는 HELM(Holistic Evaluation of Language Models)의 결과에 따르면 Cohere 모델이 가장 뛰어난 성능을 제공

Cohere 모델은 파라미터가 52B(520억개 파라미터)로 구성되고, OpenAI의 다빈치 모델은 178B(1,780억개) 파라미터를 갖음

Cohere 모델이 OpenAI 모델에 비교하여 모델 크기가 작으면서 뛰어난 성능 제공

작은 LLM 모델이 갖는 강점
- 빠른 추론, 빠른 성능 제공
- 작은 모델을 Foundation AI 모델 (Base Model)로 추가학습 시키는데 투입되는 GPU 수량과 시간을 줄임
- 운영 비용 절감 (Production에 투입 GPU 수량 감소)

Cohere 모델의 특성은 엔터프라이즈를 위한 Generative AI에 적합한 가격, 비용, 성능 효율성 제공

Generative AI for enterprise—offered through Oracle

Company	Model	Model type	Mean win rate
cohere	**Cohere Command (52B)**	**Command**	**93.0%**
OpenAI	Davinci Instruct 002	Command	93.0%
OpenAI	Davinci Instruct 003	Command	89.8%
NVIDIA Microsoft	TNLG v2 (530B) *not publicly available or viable to serve given size*	Base	85.5%
ANTHROPIC	Anthropic v4 (52B)	Command	84.2%
AI21labs	J1 Grande v2 (17B)	Command	80.6%
ALEPH ALPHA	Luminous Supreme (70B)	Command	78.3%
cohere	**Cohere XL (52B)**	**Base**	**74%**
Meta	OPT (175B)	Base	67.8%
OpenAI	GPT-3 Davinci (175B)	Base	62.8%
AI21labs	J1-Jumbo (178B)	Base	59.2%
ALEPH ALPHA	Luminous Extended (30B)	Command	58.2%
Hugging Face	BLOOM (176B)	Base	52.9%

Cohere delivers top-tier LLM performance, outperforming peers in independent LLM benchmarks

(not included: GPT-4 from OpenAI)

Cohere's Command model is ranked very highly in HELM, ***while being more efficient*** (52B parameters compared to GPT-3 with 175B parameters) and ***more easily customized***

RAG 란 무엇일까?

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 작성해 주세요

지시

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

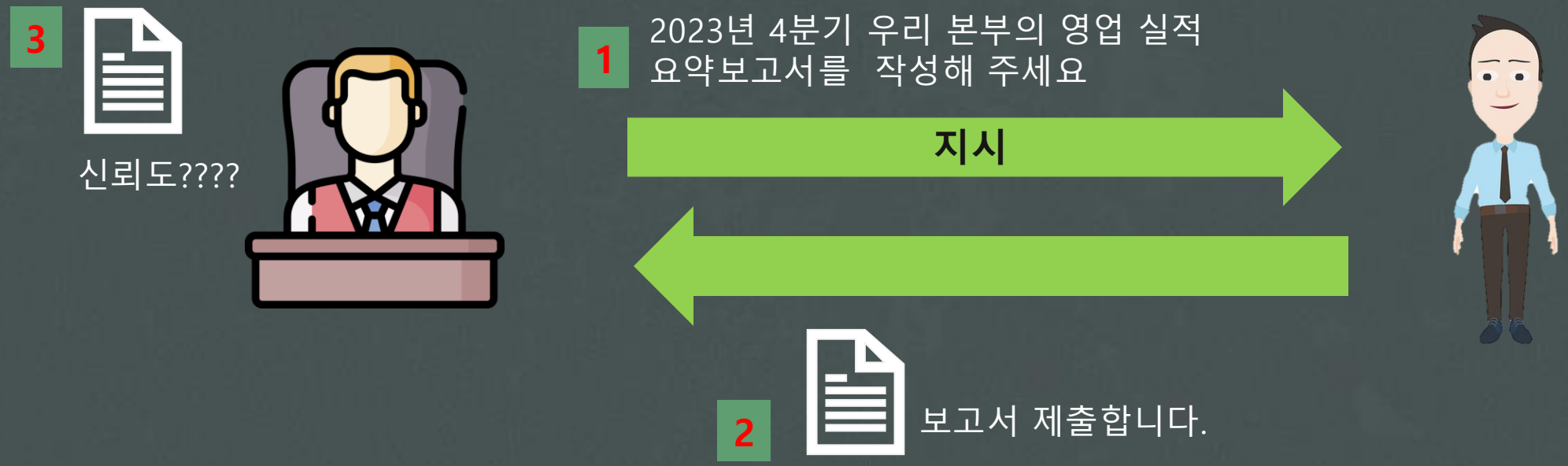
1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 작성해 주세요

지시

2 보고서 제출합니다.

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안



RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

3

이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

3 이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

4 보고서 제출합니다.

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

R
Retrieve

3

이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

4 보고서 제출합니다.

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

R
Retrieve

3

이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

A
Augmented

4 보고서 제출합니다.

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

R
Retrieve

3

이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

A
Augmented

4 보고서 제출합니다.

G
Generation

RAG (Retrieval Augmented Generation)

[예시] 현실에서의 업무 지시: 신뢰성 향상 방안

1 2023년 4분기 우리 본부의 영업 실적 요약보고서를 조이나님 에게 지시해 주세요. 관련 문서도 함께 제공해 주세요

2 관련 문서 찾기

R
Retrieve

OpenSearch

Database

Application Server

3

이 문서를 참조하여 지난 분기 본부 영업 실적 요약보고서 팀장님에게 제출하세요

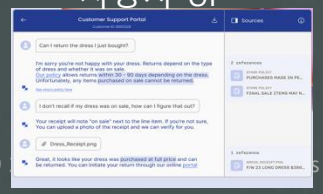
A
Augmented

4 보고서 제출합니다.

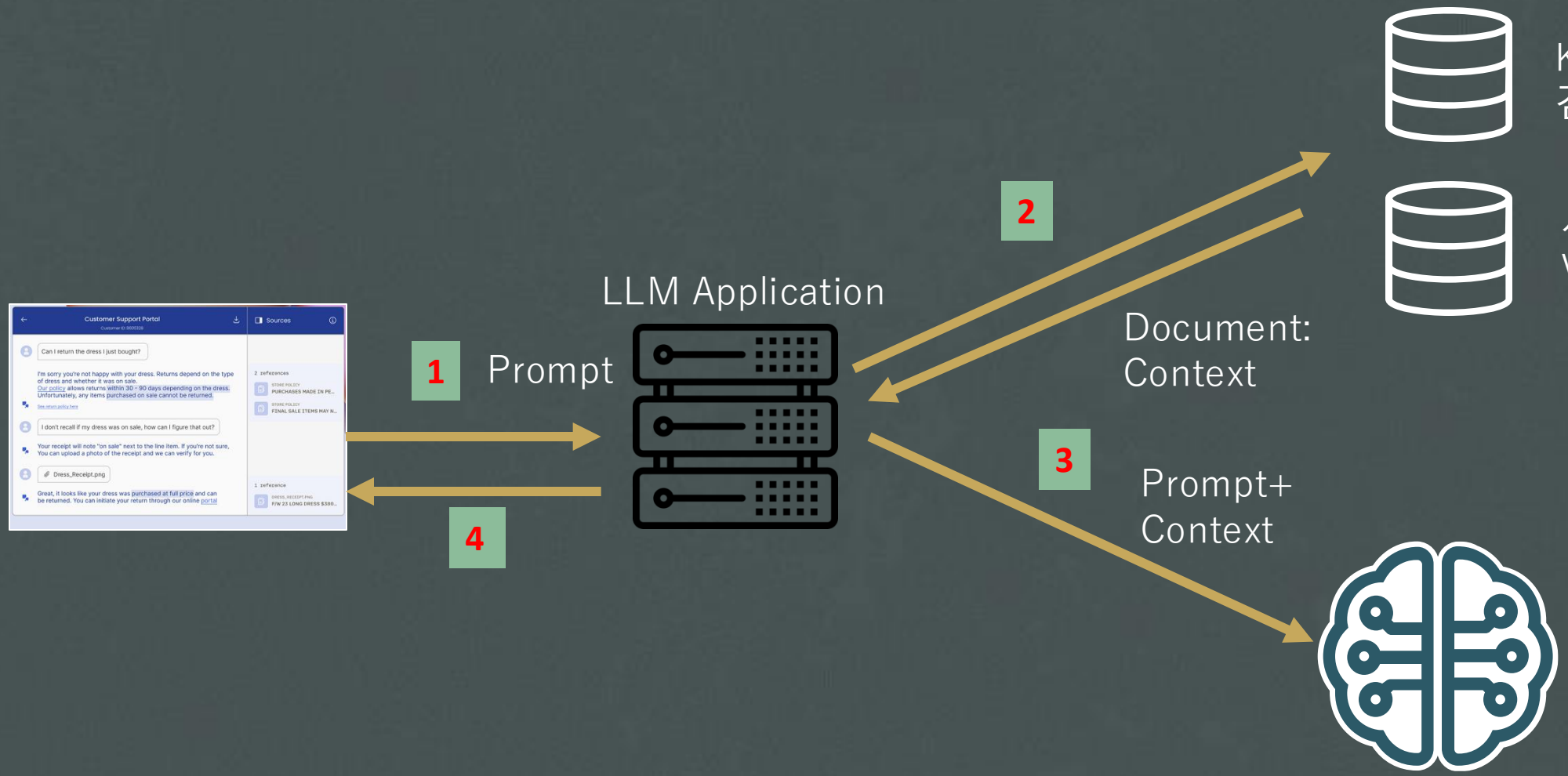
G
Generation

LLM(Large language Model)

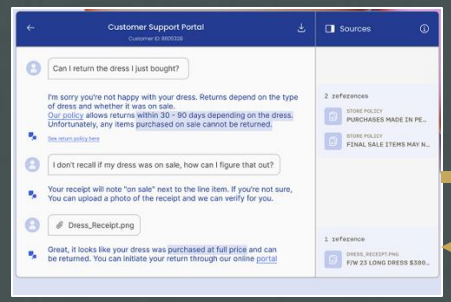
사용자 UI



RAG(Retrieval Augmented Generation) 아키텍처



RAG(Retrieval Augmented Generation) 아키텍처



LLM Application

1 Prompt

2 Document: Context

3 Prompt+ Context

4

R
Retrieve

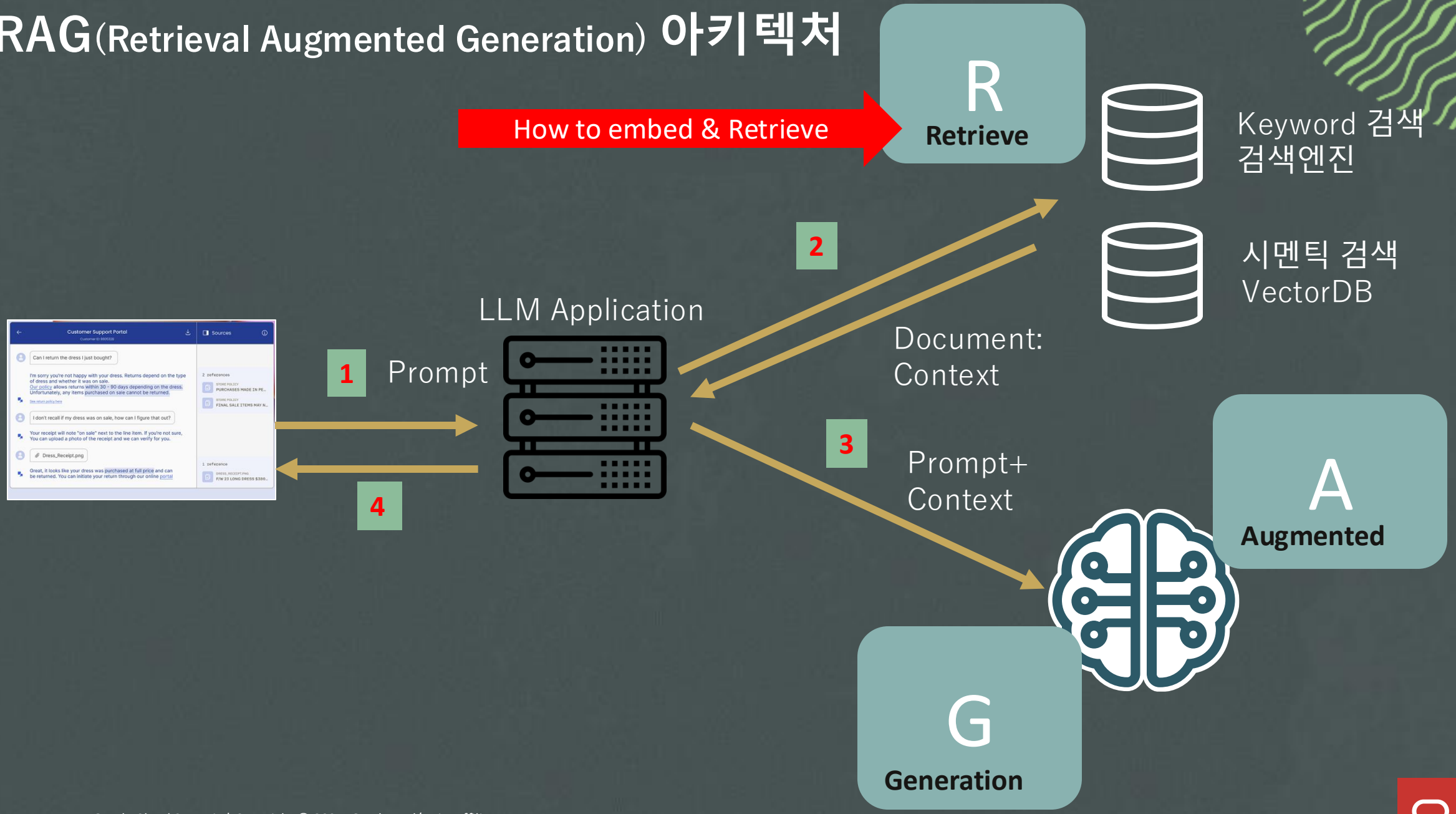
Keyword 검색
검색엔진

시멘틱 검색
VectorDB

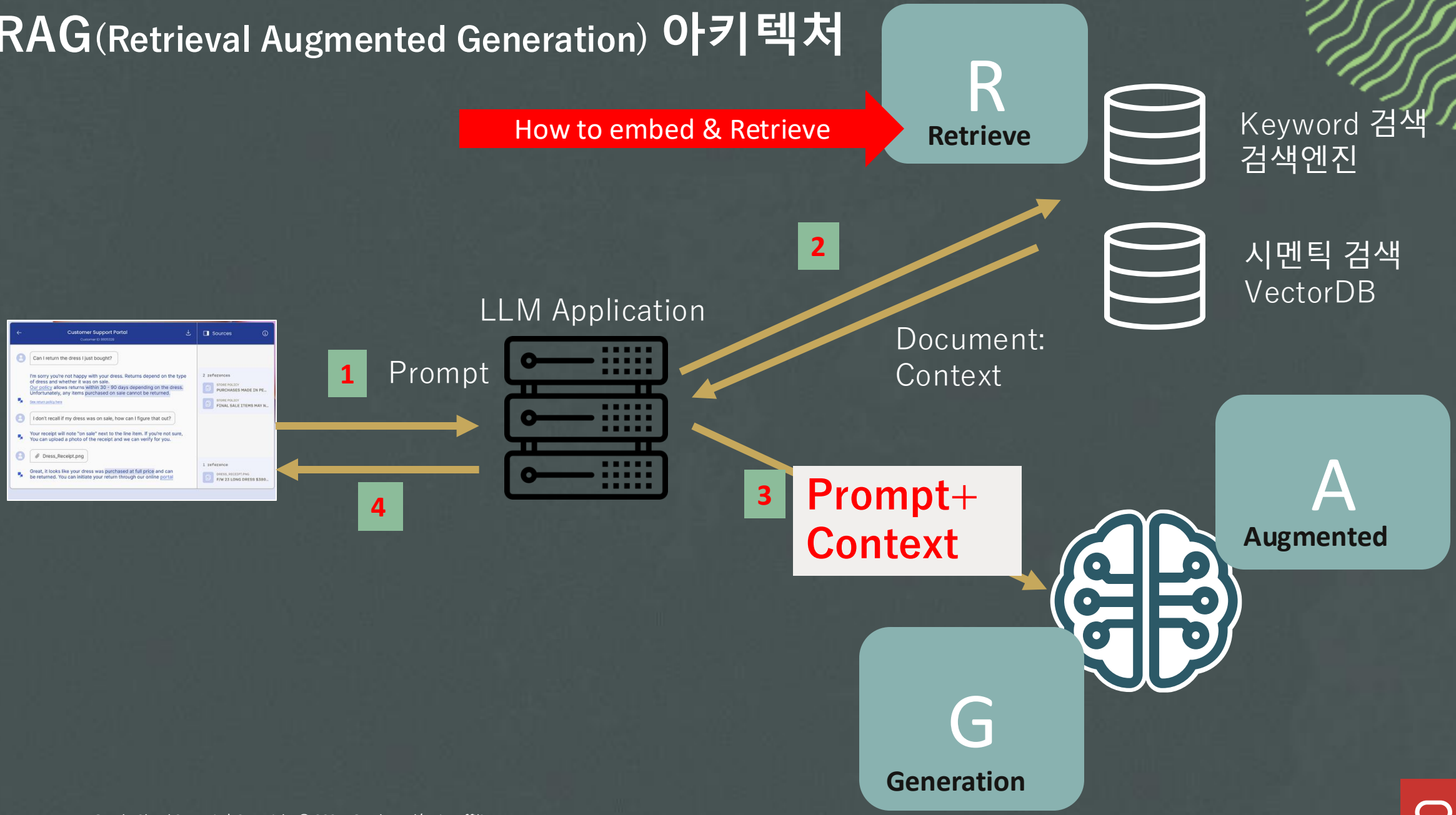
A
Augmented

G
Generation

RAG(Retrieval Augmented Generation) 아키텍처

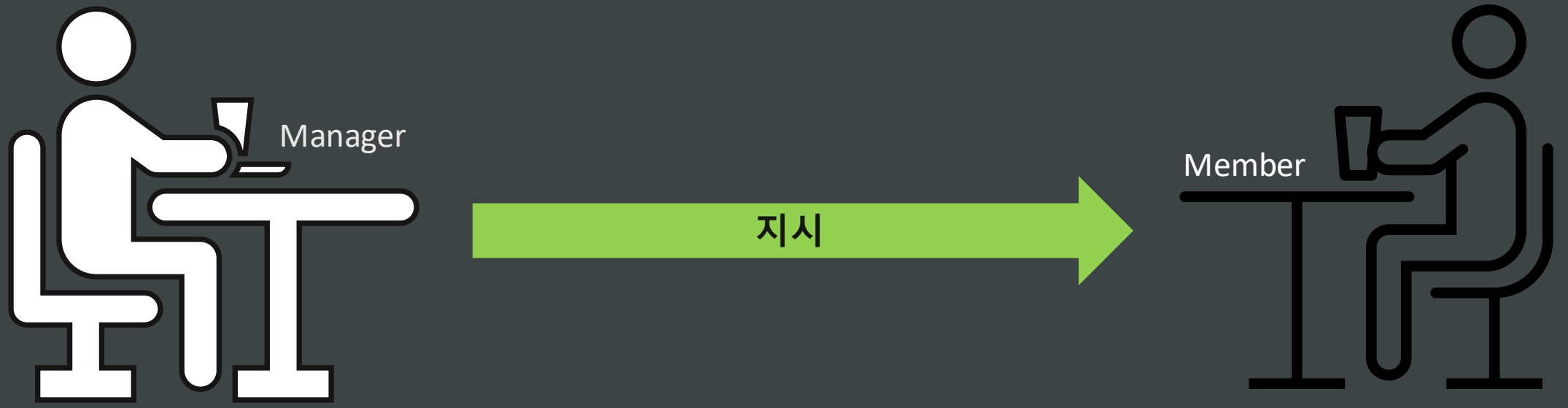


RAG(Retrieval Augmented Generation) 아키텍처



인간 커뮤니케이션

W5H1, 육하원칙



인간 커뮤니케이션

W5H1, 육하원칙

Who 누가

When 언제

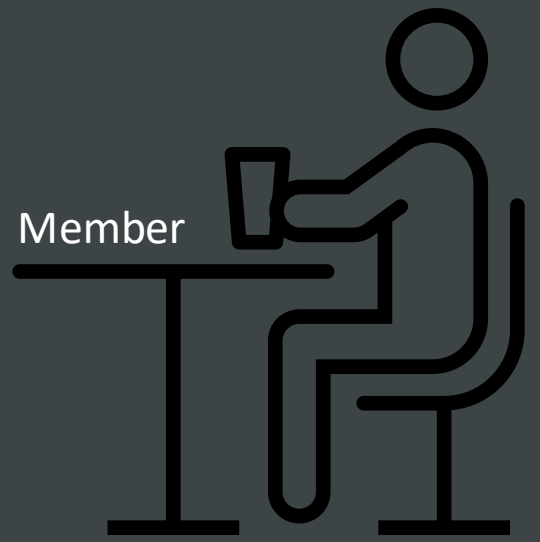
Where 어디서

지시

How 어떻게

What 무엇을

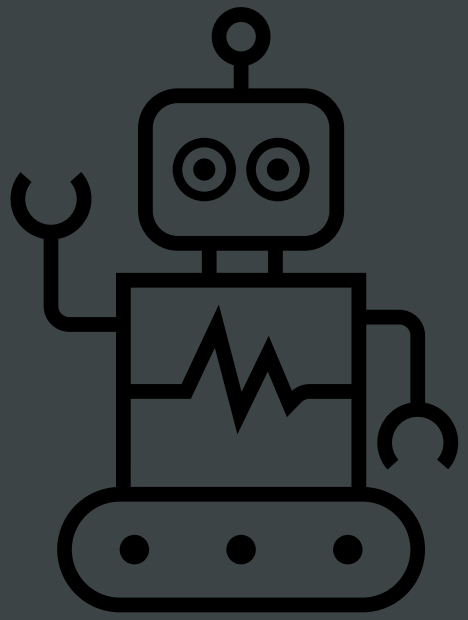
Why 왜



LLM과 커뮤니케이션

Prompt Engineering

Prompt



Generative AI (LLM, Large Language Model)

LLM과 커뮤니케이션

Prompt Engineering

Role 역할

Instruction 지시

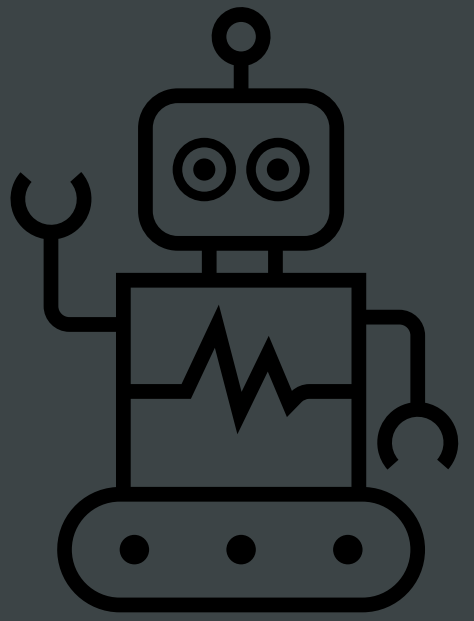
Context 문맥

Prompt

Question 질문

Few Shot 예시

Output 출력 형태



Generative AI (LLM, Large Language Model)

LLM과 커뮤니케이션

Prompt Template 설정

Prompt Engineering

Role 역할

Instruction 지시

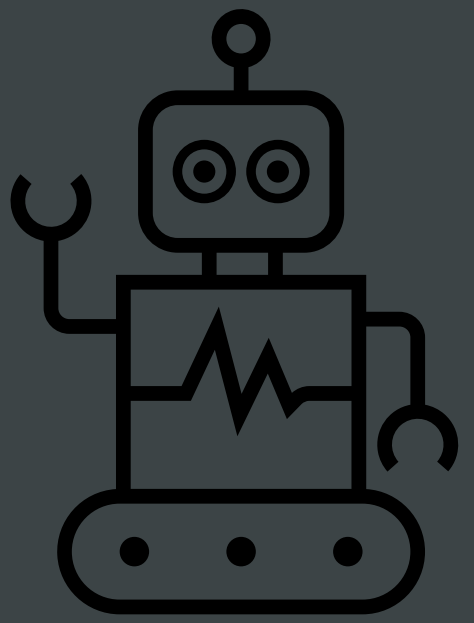
Context 문맥

Prompt

Question 질문

Few Shot 예시

Output 출력 형태



Generative AI (LLM, Large Language Model)

LLM과 커뮤니케이션

Prompt Template 설정

사용자 입력 정보 설정
- Runtime

Prompt Engineering

Role 역할

Instruction 지시

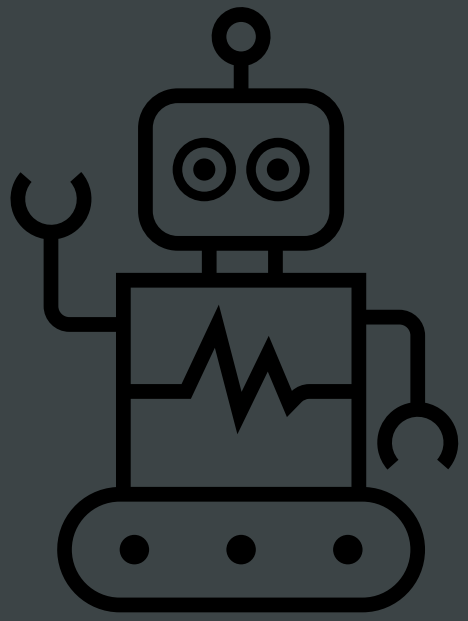
Context 문맥

Prompt

Question 질문

Few Shot 예시

Output 출력 형태



Generative AI (LLM, Large Language Model)

LLM과 커뮤니케이션

Prompt Template 설정

사용자 입력 정보 설정
- Runtime

Knowledge 조회 설정
- Runtime

Prompt Engineering

Role 역할

Instruction 지시

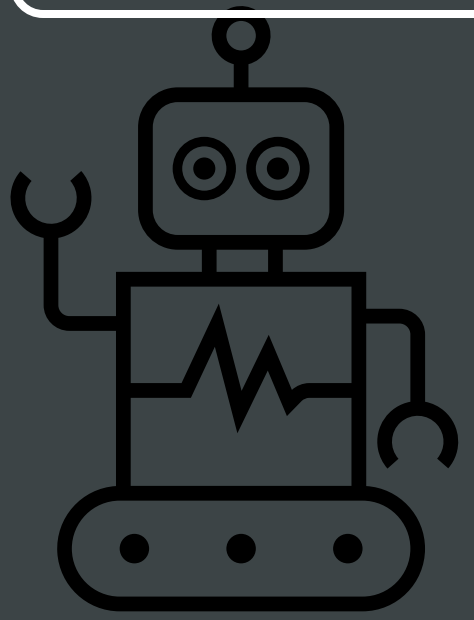
Context 문맥

Prompt

Question 질문

Few Shot 예시

Output 출력 형태



Generative AI (LLM, Large Language Model)

Vector Embedding: 비정형 데이터에 대한 의미(Semantic)용 Vector 변환

AI 벡터 검색 비정형 데이터 활용하기

벡터는 차원이라고 하는 숫자의 시퀀스로, 데이터의 중요한 **"특징"**을 포착하는 데 사용됩니다.

벡터는 기본 단어나 픽셀이 아닌 데이터의 **의미적 내용**을 나타냅니다.

벡터는 딥러닝 임베딩 모델을 사용하여 생성됩니다.

LLM과 커뮤니케이션

Prompt Engineering

Prompt Template 설정

사용자 입력 정보 설정
- Runtime

Knowledge 조회 설정
- Runtime

Role 역할

Instruction 지시

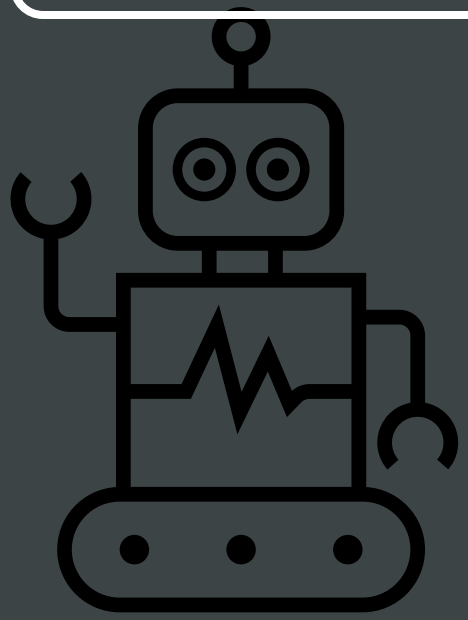
Context 문맥

Prompt

Question 질문

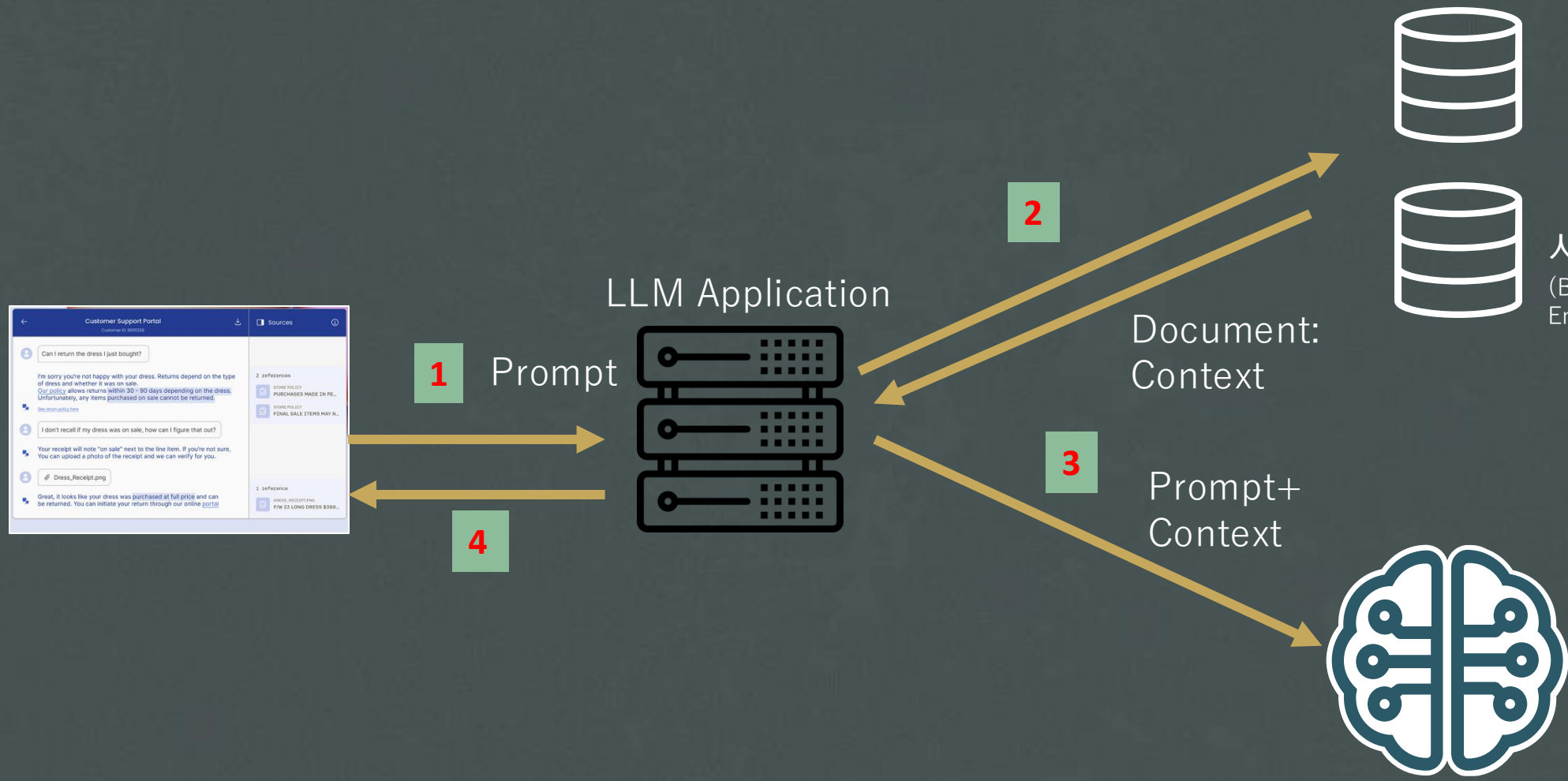
Few Shot 예시

Output 출력 형태



Generative AI (LLM, Large Language Model)

RAG(Retrieval Augmented Generation) 아키텍처

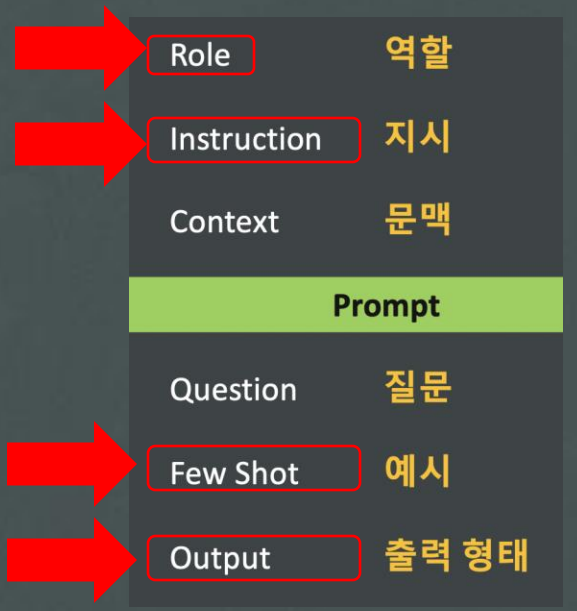


RAG(Retrieval Augmented Generation) 아키텍처+Prompt Engineering

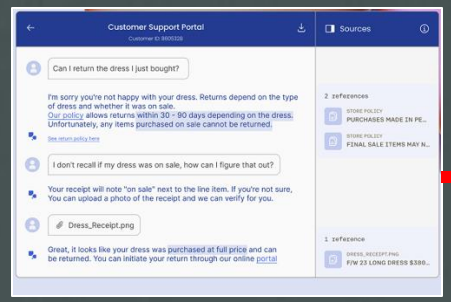
LLM Application

Restful API

Template Level 초기 설정



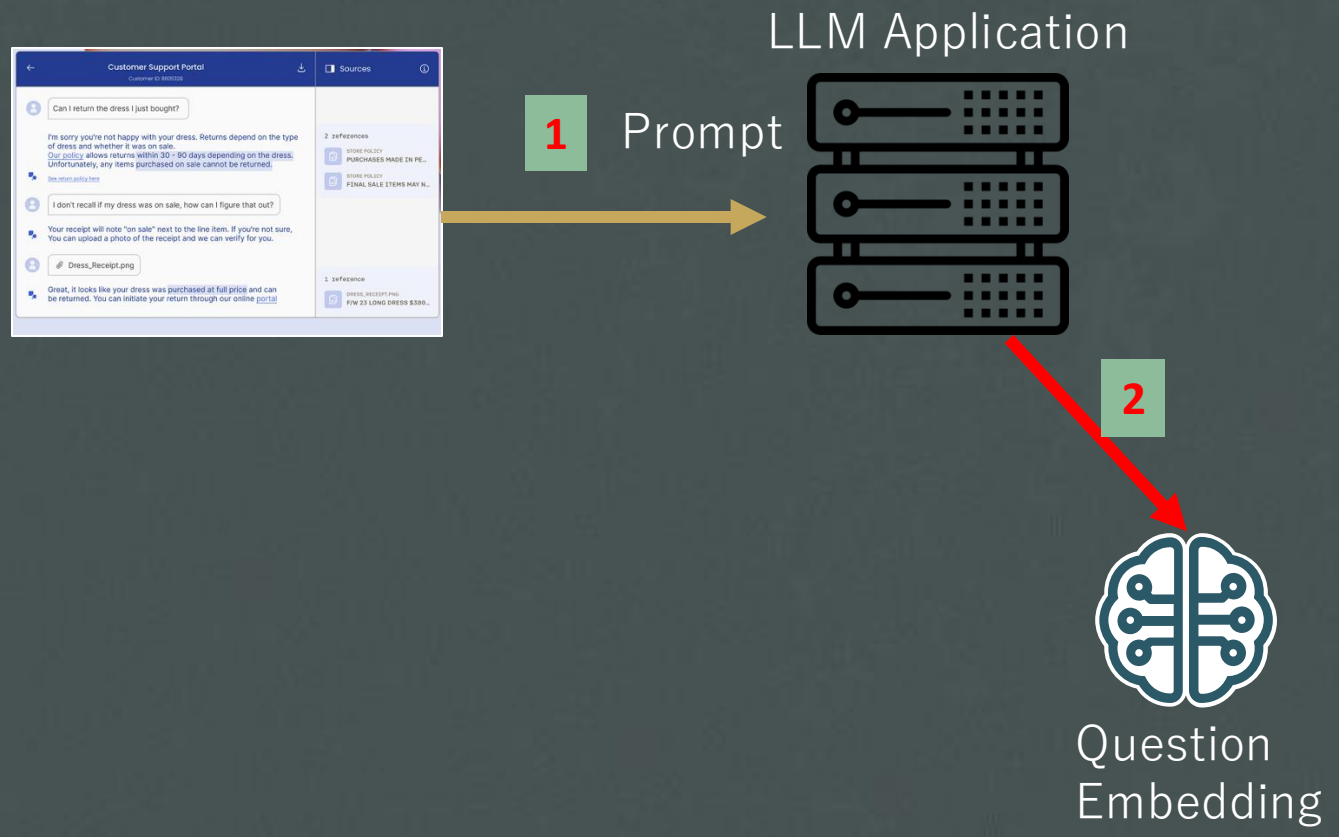
RAG(Retrieval Augmented Generation) 아키텍처+Prompt Engineering



1 Prompt

Role	역할
Instruction	지시
Context	문맥
Prompt	
Question	질문
Few Shot	예시
Output	출력 형태

RAG(Retrieval Augmented Generation) 아키텍처



RAG(Retrieval Augmented Generation) 아키텍처

OCI Cache
: Managed Redis
→ History of Prompt

2-0

3

Keyword 검색
검색엔진

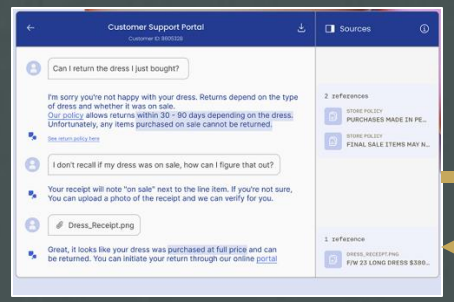
시멘틱 검색
VectorDB

LLM Application

LangChain
LLamaIndex
Spring AI

Document:
Context

1 Prompt



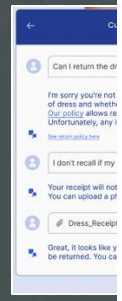
5

2

Question
Embedding

Role	역할
Instruction	지시
Context	문맥
Prompt	
Question	질문
Few Shot	예시
Output	출력 형태

RAG



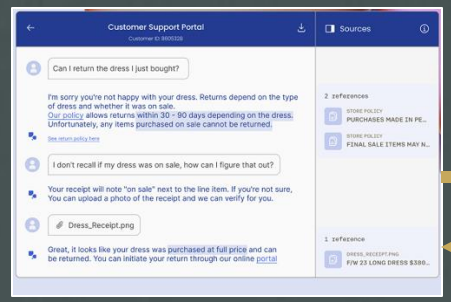
```
template = """당신은 한국은행에서 출간한 금융 용어를 설명해주는 'K금융이'입니다.
# Instrcution
1. 주어진 질문을 카테고리로 분류해줘
- 금융, 연애인, 문화, 정치, 인종차별, 지역감정, 성정체성, 범죄, 일상대화
2. 질문이 금융 및 일상대화가 아니라면 다음과 같이 답변해줘
- 저는 금융 용어 챗봇입니다. 금융 관련 질문이 아닌 것은 대답할 수 없습니다.
- 답변을 하지 않은 이유에 애해서 진물 분류 tag와 함께 설명해줘
3. 주어진 검색 결과를 바탕으로 답변하세요. 검색 결과에 관련 내용이 없다면 답변하지 마세요.
4. 사용자는 금융 입문자 입니다. 최대한 쉽고 간결하게 설명해 주세요.
5. 질문에 대한 요약을 먼저 출력
6. 답변은 리스트 형태로 구조화하여 작성할 것
#예시:
<example 1: 질문이 금융 및 일상대화가 아니라면>
질문: 나는 보수주의자야? 진보 주의자야?
출력:
- 답변: 죄송합니다. 저는 금융 용어 챗봇입니다. 금융 관련 질문이 아닌 것은 대답할 수 없습니다.
- 질문유형: 정치
<example 2: 질문이 일상대화라면>
질문: 안녕하세요!
출력:
- 답변: 안녕하세요. 반갑습니다.
- 질문유형: 정치
<example 3: 질문이 금융 관련이라면>
질문: 비트코인데 대해서 알려줘
출력:
비트코인에대해서 문의하겼습니다.
- 비트코인은 전화 화폐의 일종입니다.
- 질문유형: 금융

# 검색결과
{context}

# 질문
{question}
"""

prompt = PromptTemplate.from_template(template)
chain = (  {"context": retriever, "question": RunnablePassthrough()} | prompt | oci_llm)
```

RAG(Retrieval Augmented Generation) 아키텍처



1 Prompt

5

3

Document:
Context

Keyword 검색
검색엔진

시멘틱 검색
VectorDB

4 Prompt+

Role	역할
Instruction	지시
Context	문맥
Prompt	
Question	질문
Few Shot	예시
Output	출력 형태

2

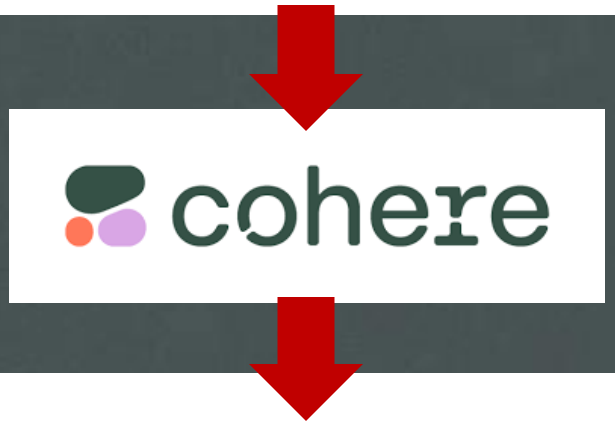
Question
Embedding

Command R+: RAG 및 추적성 특화 모델

RAG를 위한 Fine-Tuning & Citation 지원

Prompt

```
{
    "message": "Where do the tallest penguins live?”
}
```

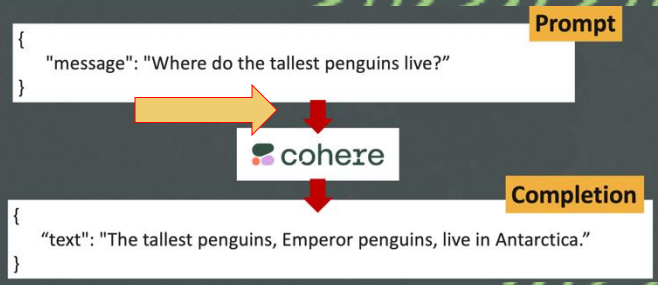


Completion

```
{
    “text": "The tallest penguins, Emperor penguins, live in Antarctica.”
}
```

Command R+: RAG 및 추적성 특화 모델

RAG를 위한 Fine-Tuning & Citation 지원




```
{
    "message": "Where do the tallest penguins live?"
}
```

RAG: Knowledge Store로 부터 Retrieve를 통한 Context 확장 (By LLM Application)

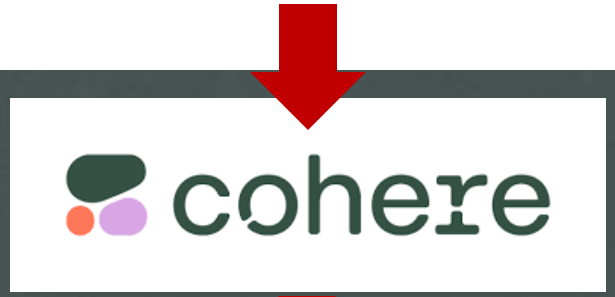
```
{ "message": "Where do the tallest penguins live?", "documents": [.
      { "title": "Tall penguins",
        "snippet": "Emperor penguins are the tallest." },
      {"title": "Penguin habitats",
        "snippet": "Emperor penguins only live in Antarctica." },
      {"title": "What are animals?",
         "snippet": "Animals are different from plants." }
   ],
}
```

Command R+: RAG 및 추적성 특화 모델

RAG를 위한 Fine-Tuning & Citation 지원

Prompt With Context

```
{ "message": "Where do the tallest penguins live?", "documents": [.
      { "title": "Tall penguins",
        "snippet": "Emperor penguins are the tallest." },
      {"title": "Penguin habitats",
        "snippet": "Emperor penguins only live in Antarctica." },
      {"title": "What are animals?",
         "snippet": "Animals are different from plants." }
   ],
}
```



+ meta Info

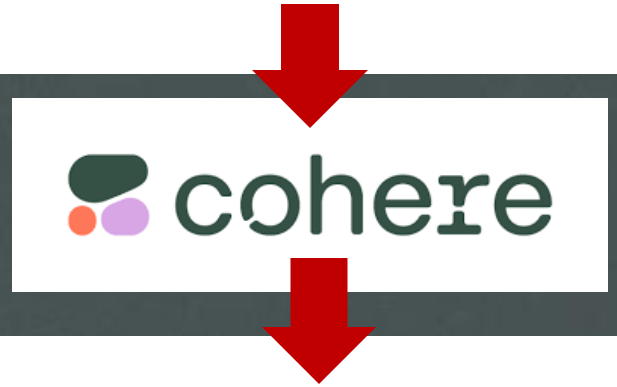
```
{
    “text": "The tallest penguins, Emperor penguins, live in Antarctica.”
}
```

Command R+: RAG 및 추적성 특화 모델

RAG를 위한 Fine-Tuning & Citation 지원

Prompt

```
{ "message": "Where do the tallest penguins live?", "documents": [.
      { "title": "Tall penguins",
        "snippet": "Emperor penguins are the tallest." },
      {"title": "Penguin habitats",
        "snippet": "Emperor penguins only live in Antarctica." },
      {"title": "What are animals?",
         "snippet": "Animals are different from plants." }
   ],
}
```



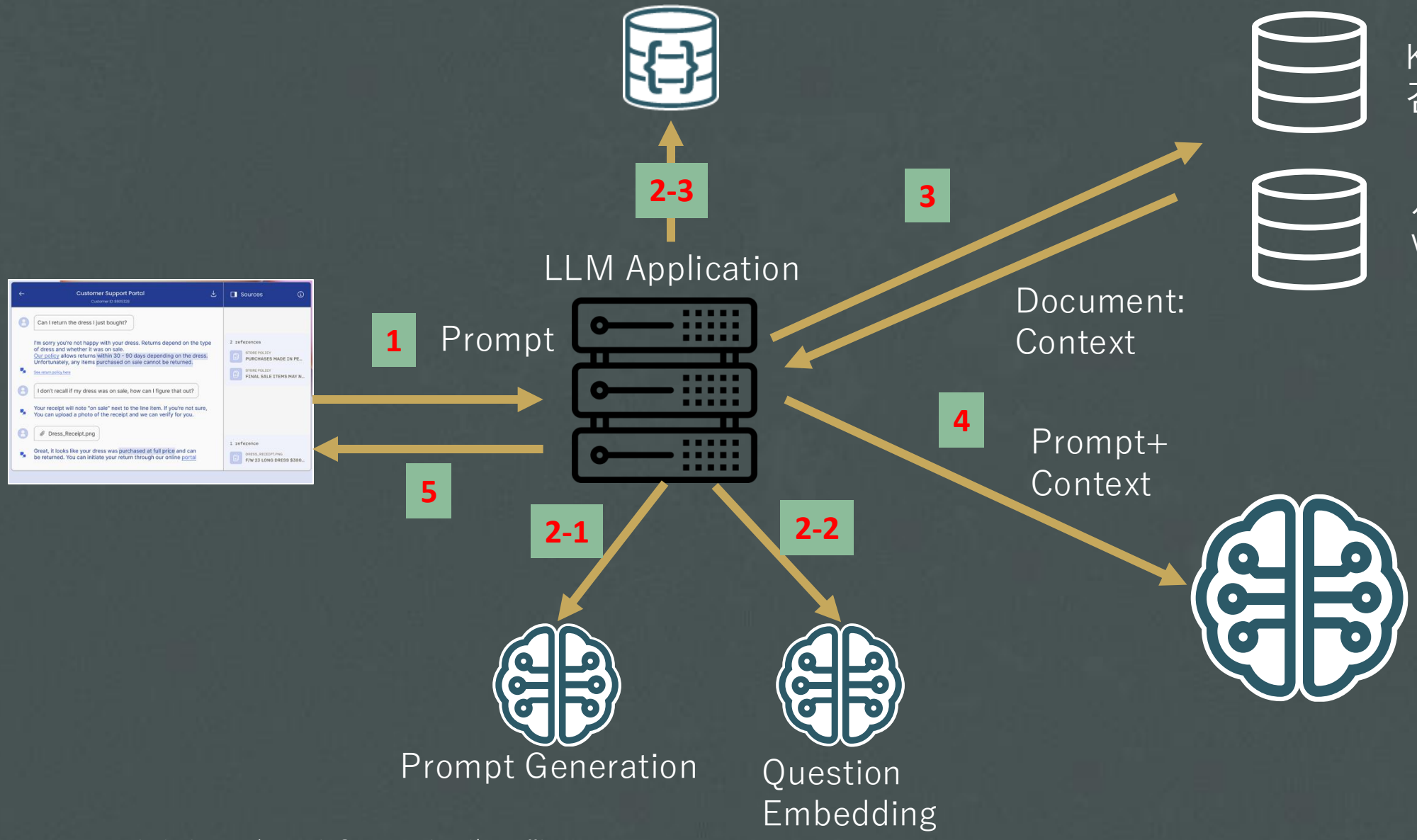
Completion

{

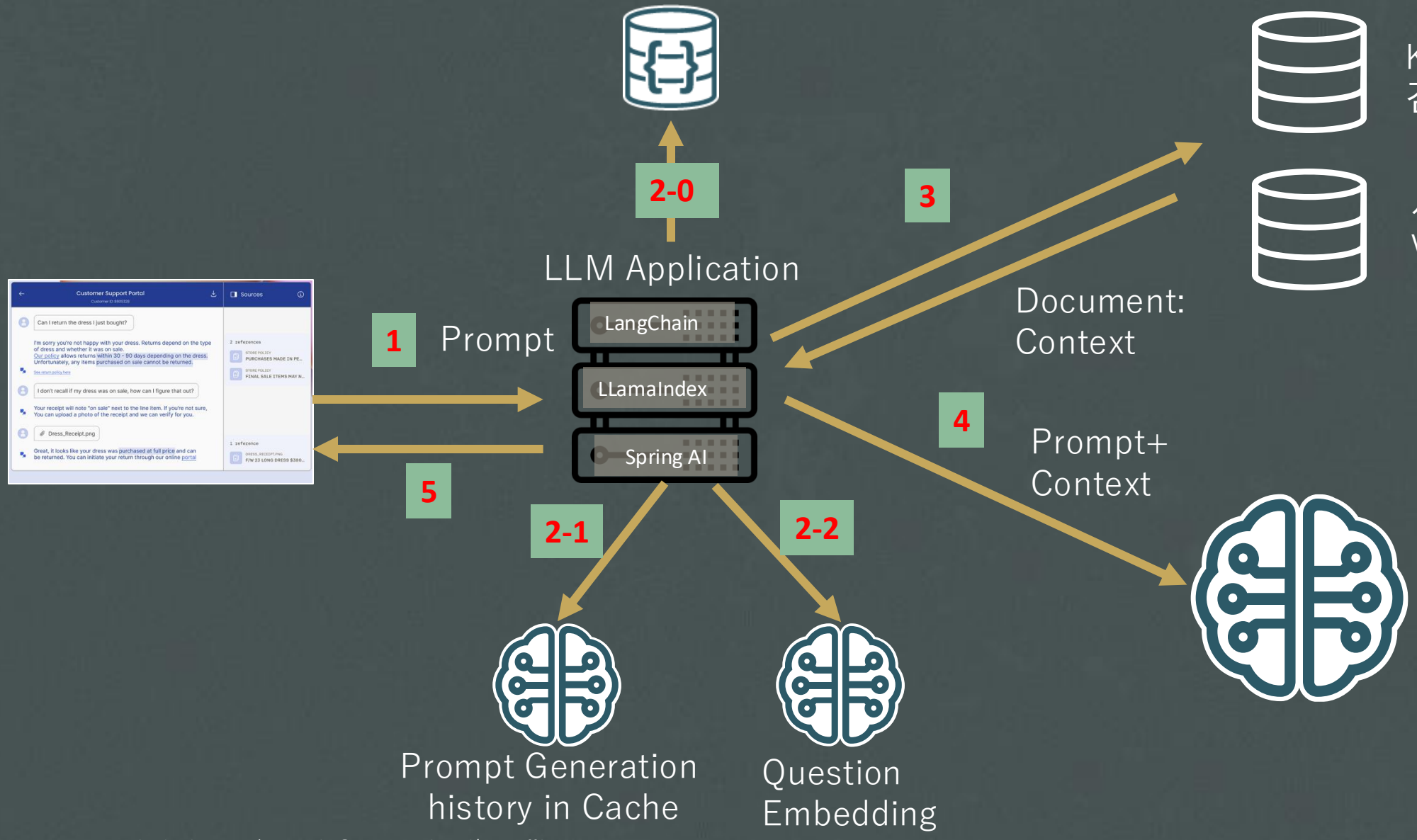
“text": ”<doc0>The tallest</doc0> <prompt>penguins</prompt>, <doc0>Emperor penguins<doc0>, live in <doc1>Antarctica</doc1>.”

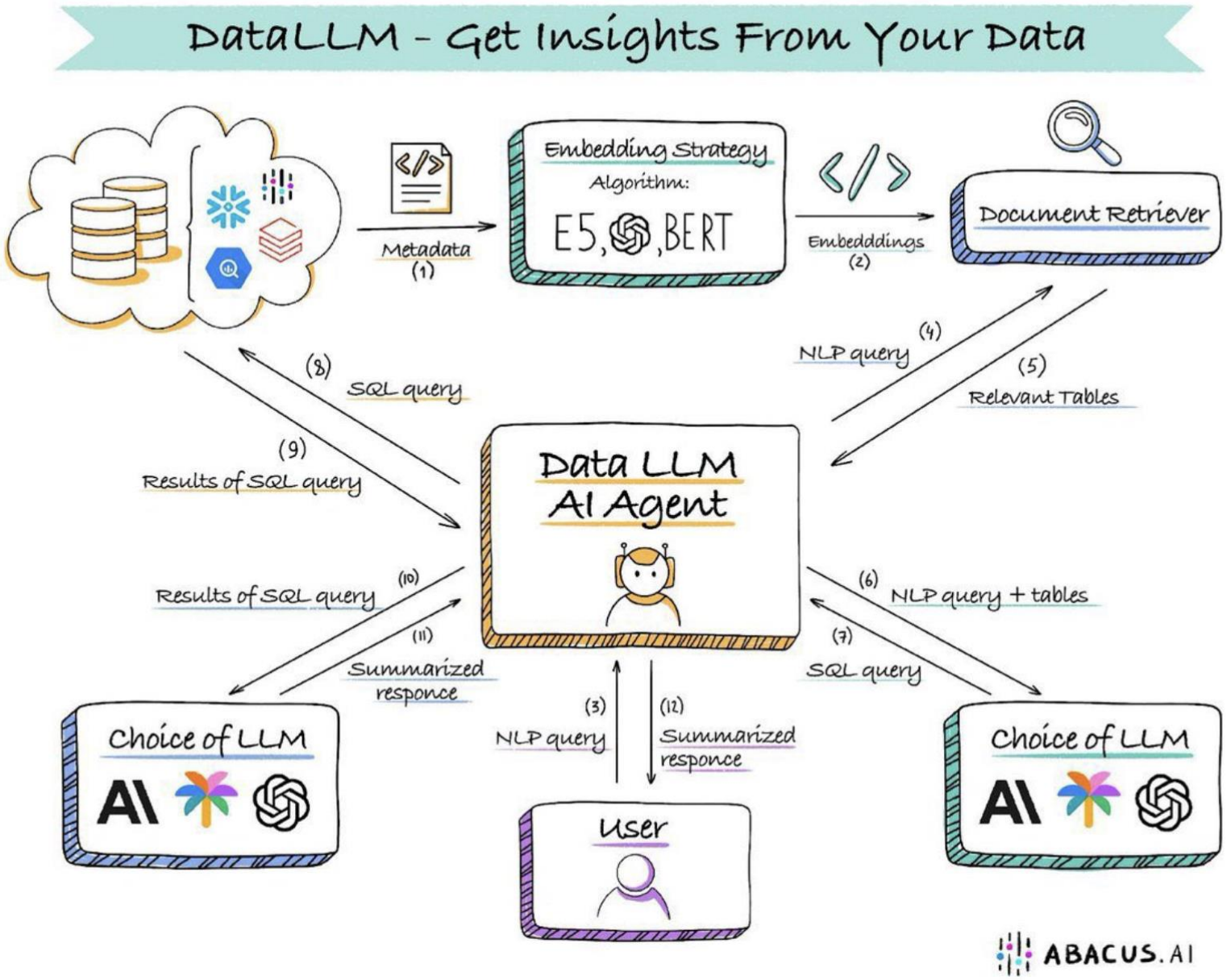
}

RAG(Retrieval Augmented Generation) 아키텍처

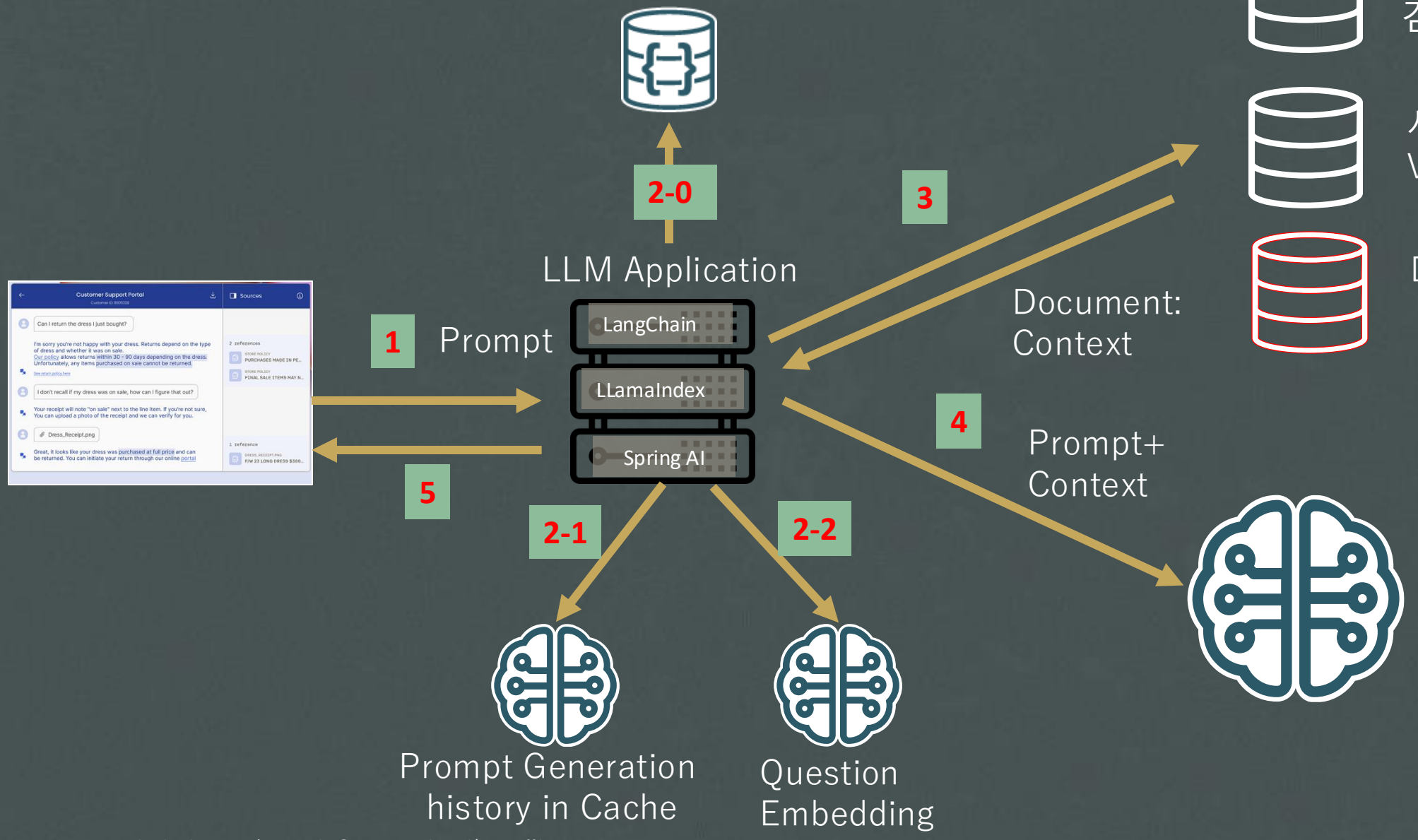


RAG(Retrieval Augmented Generation) 아키텍처

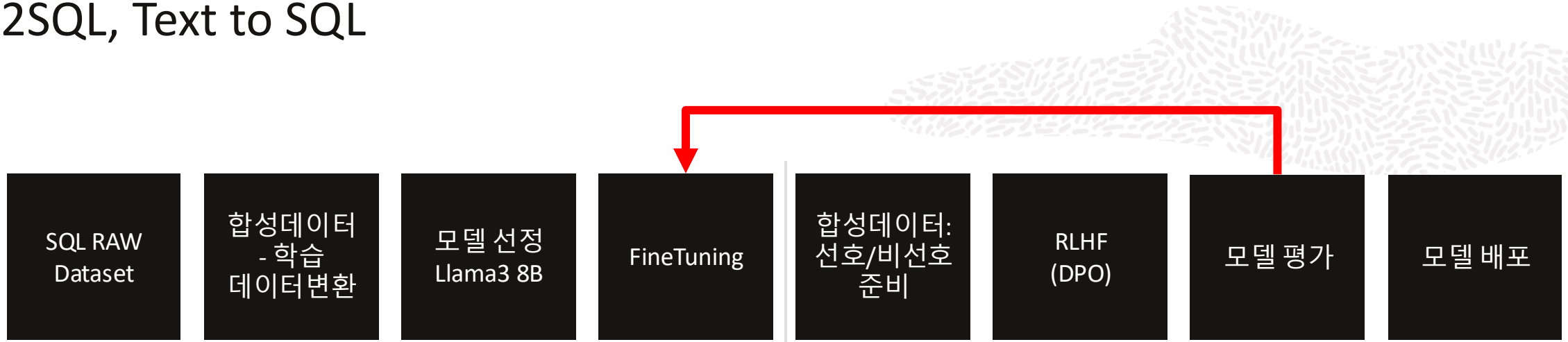




RAG(Retrieval Augmented Generation) 아키텍처:



NL2SQL, Text to SQL



NL2SQL, Text to SQL

SQL RAW Dataset | 합성데이터 - 학습 데이터변환 | 모델 선정 Llama3 8B | FineTuning | 합성데이터: 선호/비선호 준비 | RLHF (DPO) | 모델 평가 | 모델 배포

Hugging Face is way more fun with friends and colleagues! 🤗 Join an organization Dismiss this message

Datasets: Shritama/**nl2sql** like 1

Dataset card | Viewer | Files | Community 1

Split (1)
train · 10.7k rows

Search this dataset SQL Console

instruction string · *lengths* 11 – 303	input string · *lengths* 38 – 5.51k	output string · *lengths* 25 – 1.92k	text string · *lengths* 321 – 6.52k
List the number of courses in year 2008…	CREATE TABLE advisor (s_ID varchar(5), i_ID varchar(5)) CREATE TABLE prereq (…	SELECT name, COUNT(name) FROM course AS T1 JOIN teaches AS T2 ON T1.course_id =…	Below are sql tables schemas paired with instruction that describes a task. Using…
What was the result of the los angeles rams…	CREATE TABLE table_9746 ("Week" real, "Date" text, "Opponent" text, "Result"…	SELECT "Result" FROM table_9746 WHERE "Opponent" = 'los angeles rams'	Below are sql tables schemas paired with instruction that describes a task. Using…
does any airline have a JET flight between…	CREATE TABLE ground_service (city_code text, airport_code text, transport_type…	SELECT DISTINCT flight.flight_id FROM aircraft, airport_service AS…	Below are sql tables schemas paired with instruction that describes a task. Using…
is TW flight 497 766	CREATE TABLE ground_service (city_code	SELECT DISTINCT FLIGHT_0.flight_id FROM	Below are sql tables schemas paired with

NL2SQL, Text to SQL

SQL RAW Dataset

합성데이터 - 학습 데이터변환

모델 선정 Llama3 8B

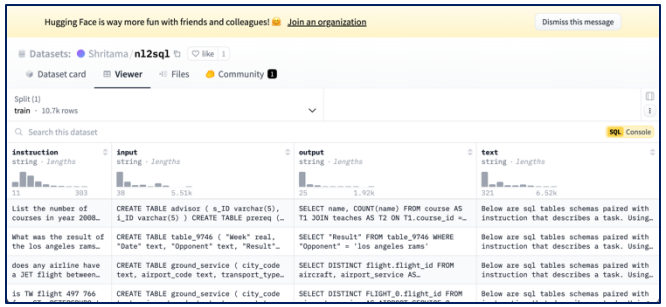
FineTuning

합성데이터: 선호/비선호 준비

RLHF (DPO)

모델 평가

모델 배포



Instruction, SQL, DDL

Instruction => 한글
ANSI SQL => 대상 DB SQL
DDL => 대상 DB DDL

NL2SQL, Text to SQL

SQL RAW Dataset

합성데이터 - 학습 데이터변환

모델 선정 Llama3 8B

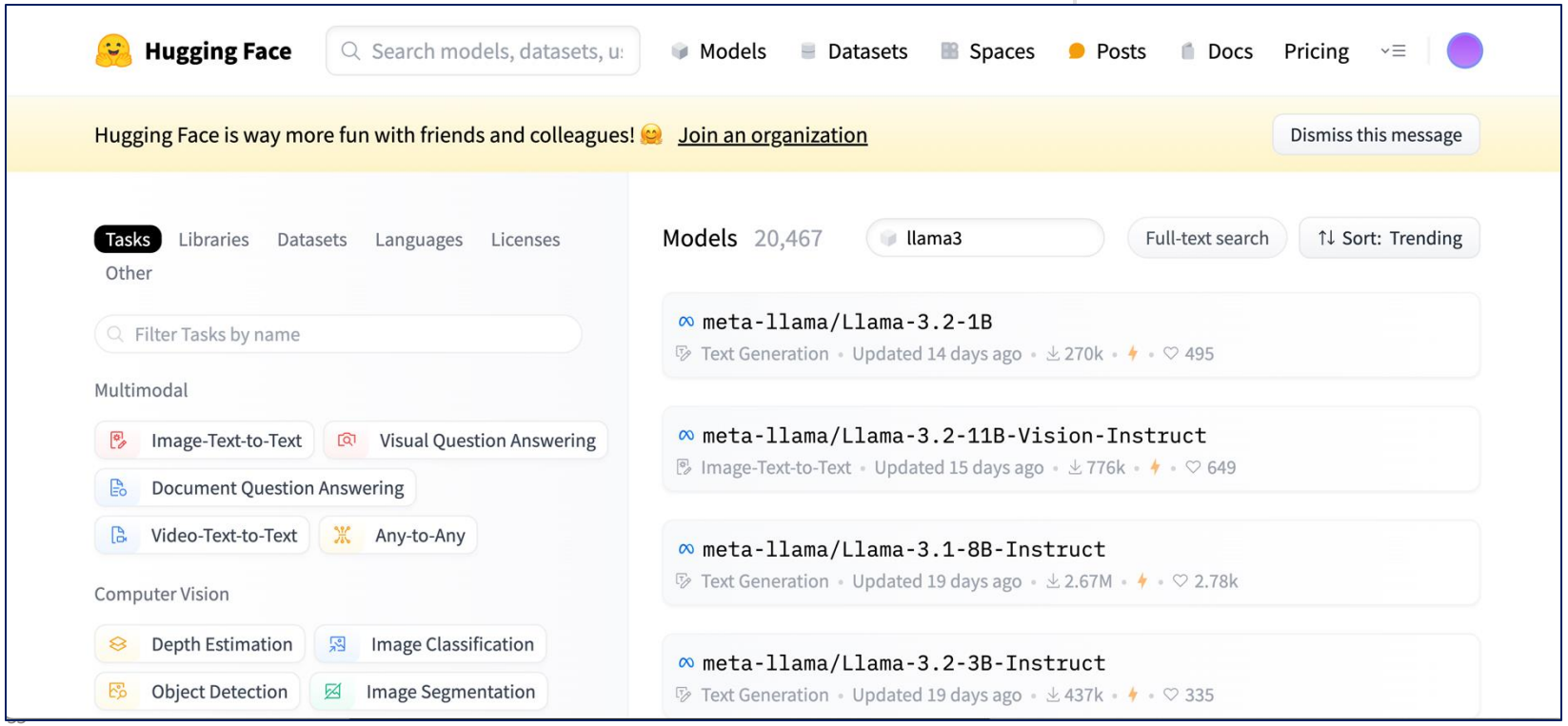
FineTuning

합성데이터: 선호/비선호 준비

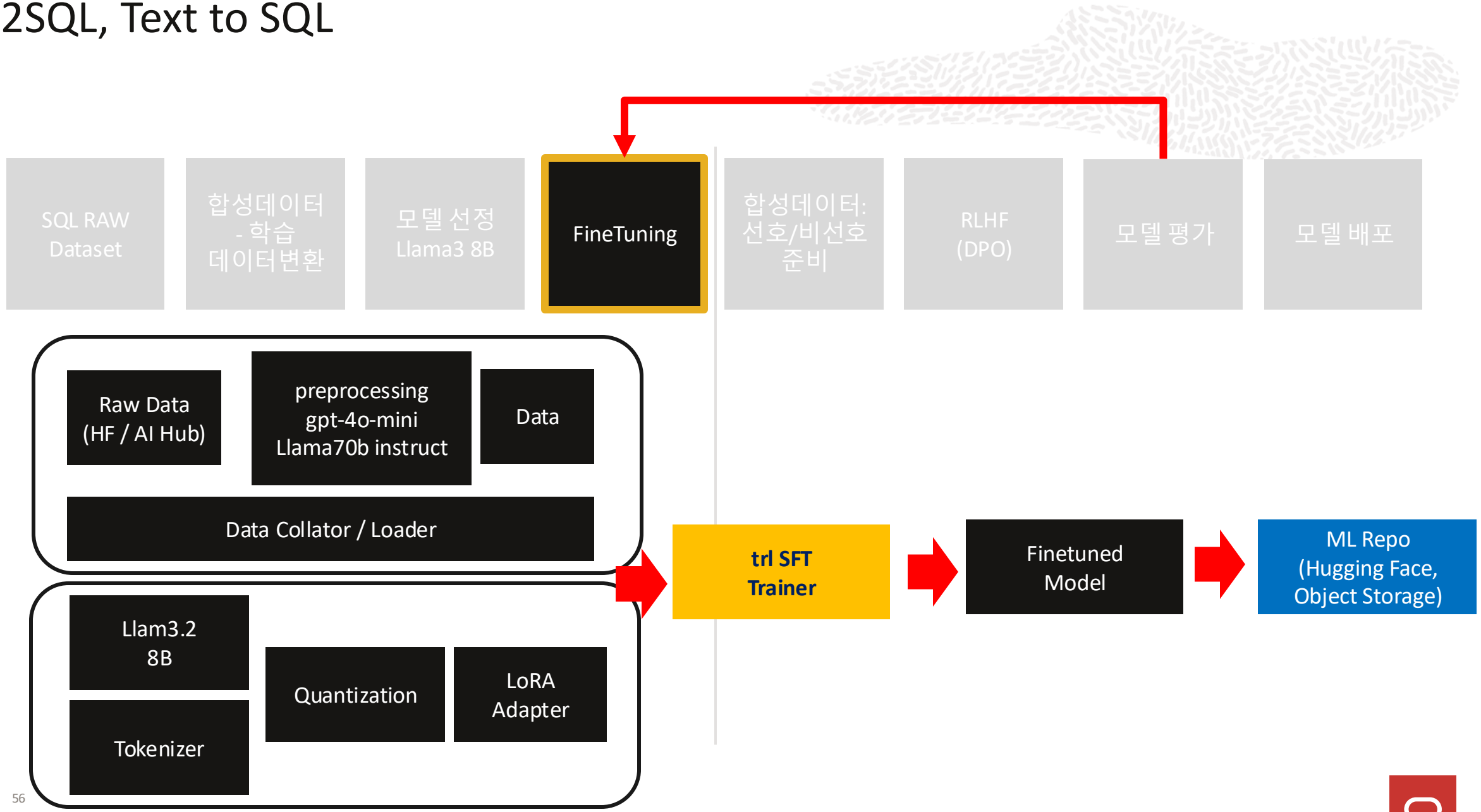
RLHF (DPO)

모델 평가

모델 배포

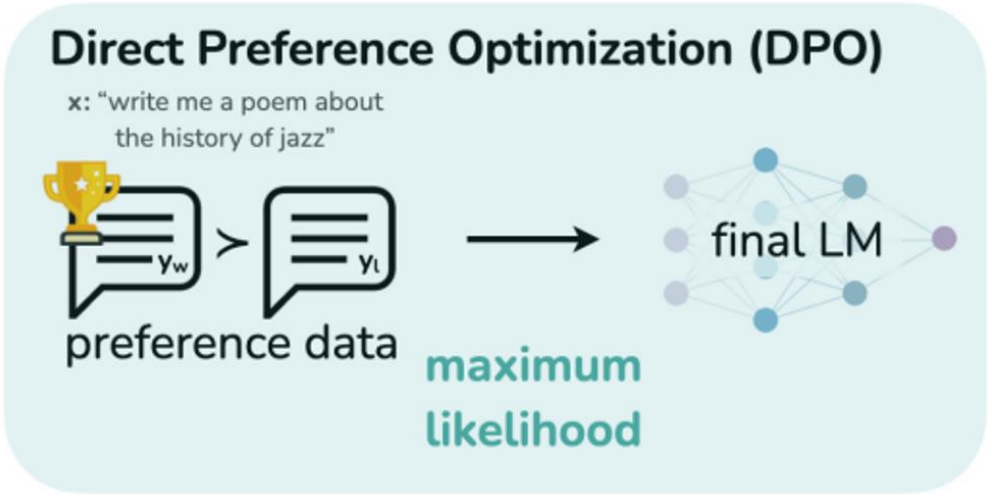


NL2SQL, Text to SQL



NL2SQL, Text to SQL

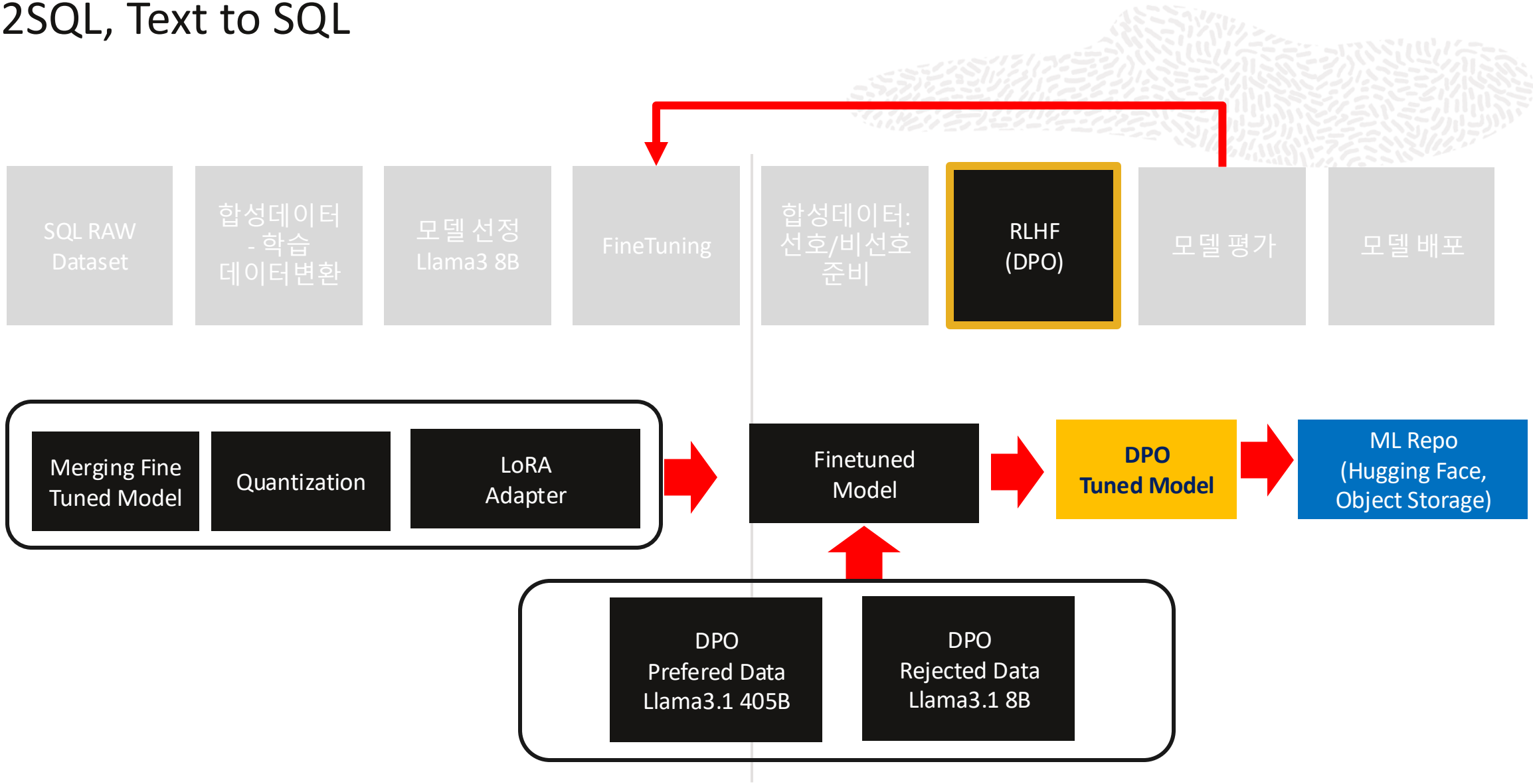
SQL RAW Dataset | 합성데이터 - 학습 데이터변환 | 모델 선정 Llama3 8B | FineTuning | 합성데이터: 선호/비선호 준비 | RLHF (DPO) | 모델 평가 | 모델 배포



Synthetic Data

선호/비선호 데이터 생성

NL2SQL, Text to SQL



NL2SQL, Text to SQL

SQL RAW Dataset

합성데이터 - 학습 데이터변환

모델 선정 Llama3 8B

FineTuning

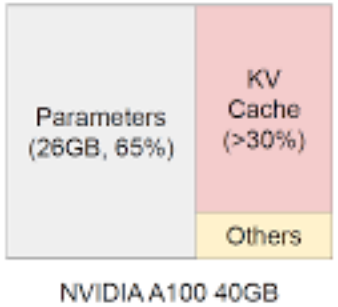
합성데이터: 선호/비선호 준비

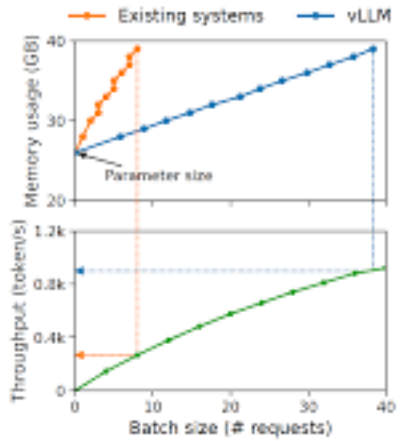
RLHF (DPO)

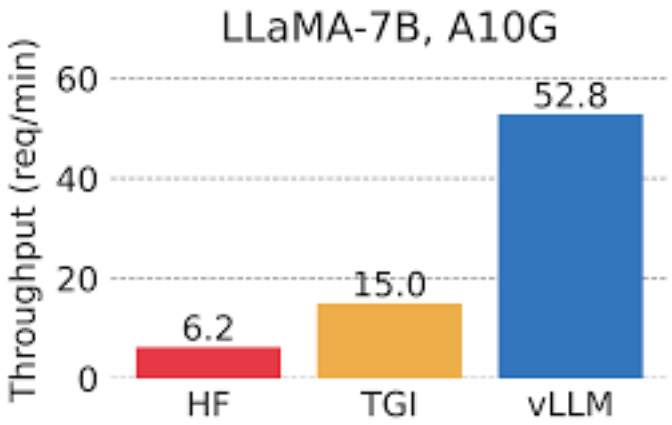
모델 평가

모델 배포

vLLM







NL2SQL, Text to SQL

SQL RAW Dataset

합성데이터 - 학습 데이터변환

모델 선정 Llama3 8B

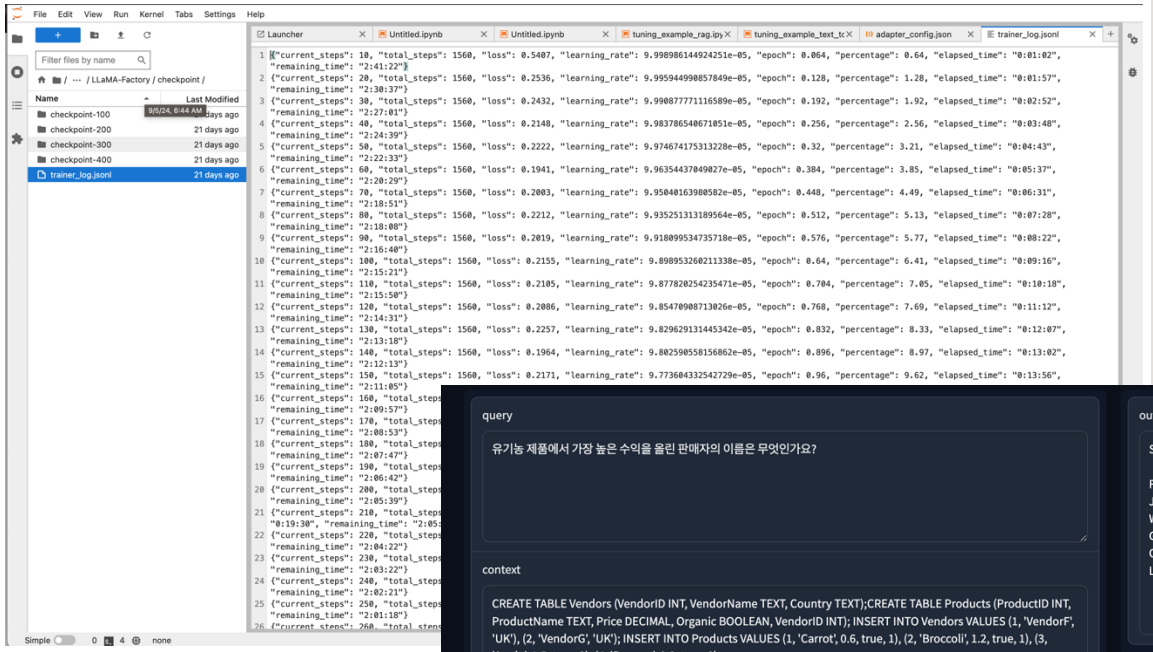
FineTuning

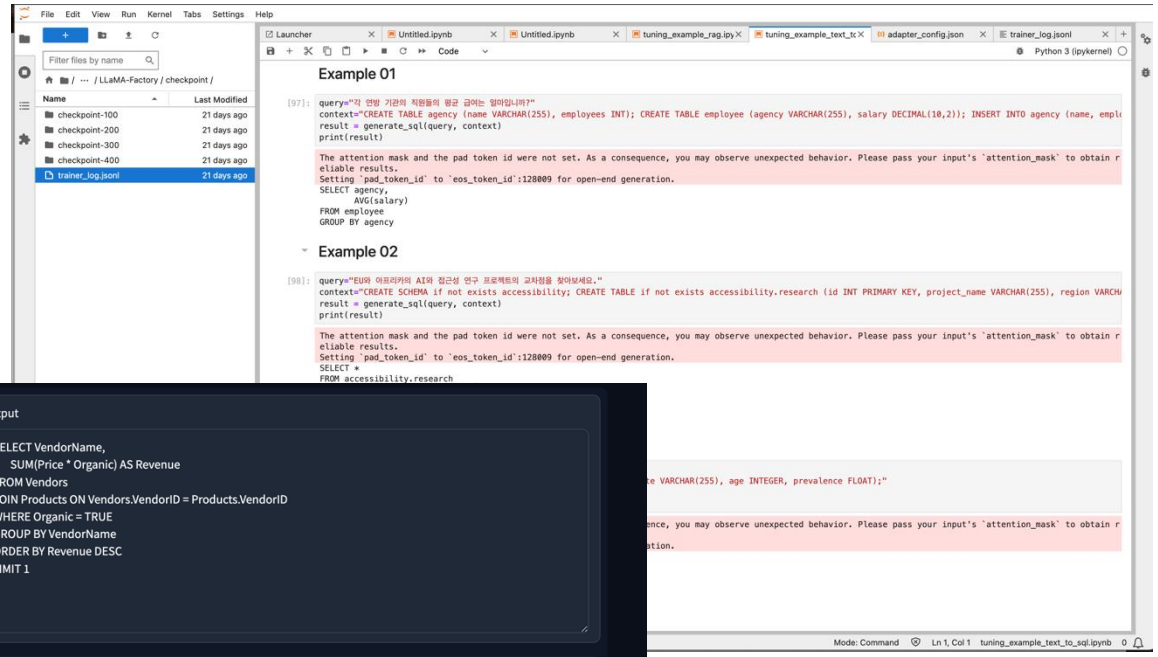
합성데이터: 선호/비선호 준비

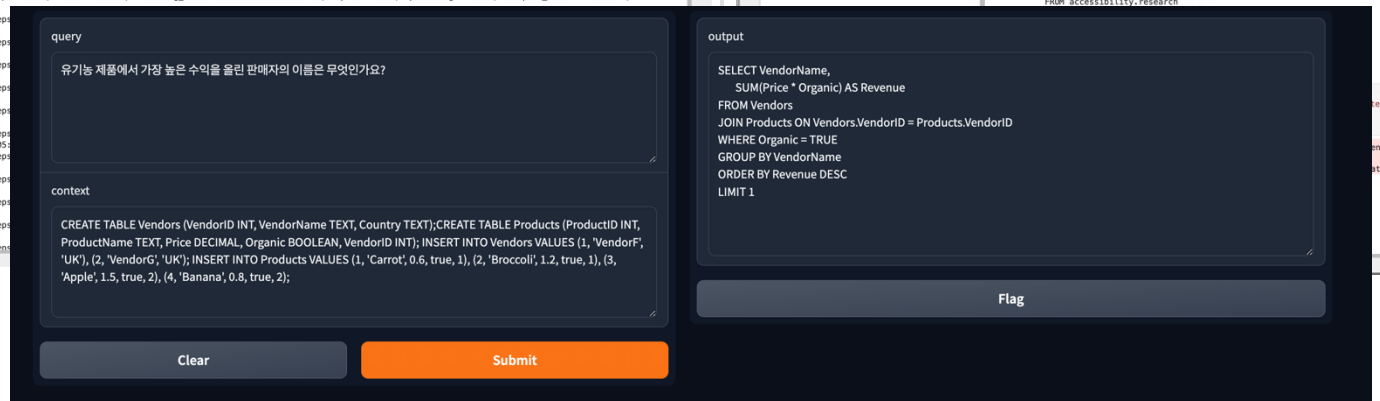
RLHF (DPO)

모델 평가

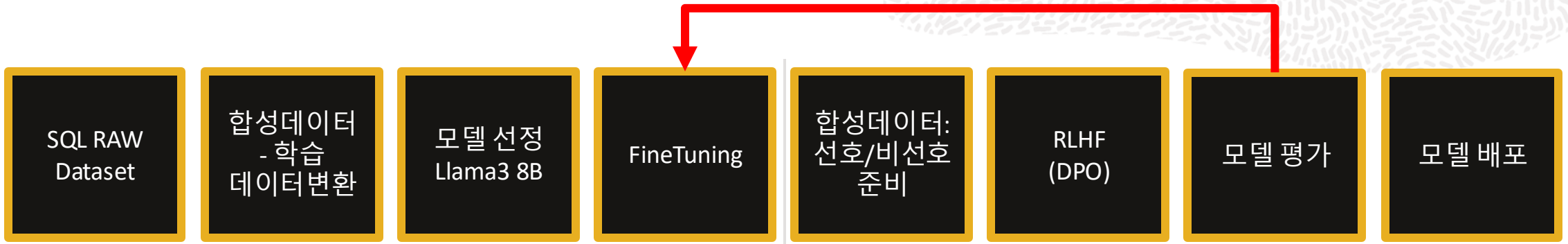
모델 배포







NL2SQL, Text to SQL



query

유기농 제품에서 가장 높은 수익을 올린 판매자의 이름은 무엇인가요?

context

CREATE TABLE Vendors (VendorID INT, VendorName TEXT, Country TEXT);CREATE TABLE Products (ProductID INT, ProductName TEXT, Price DECIMAL, Organic BOOLEAN, VendorID INT); INSERT INTO Vendors VALUES (1, 'VendorF', 'UK'), (2, 'VendorG', 'UK'); INSERT INTO Products VALUES (1, 'Carrot', 0.6, true, 1), (2, 'Broccoli', 1.2, true, 1), (3, 'Apple', 1.5, true, 2), (4, 'Banana', 0.8, true, 2);

Clear

Submit

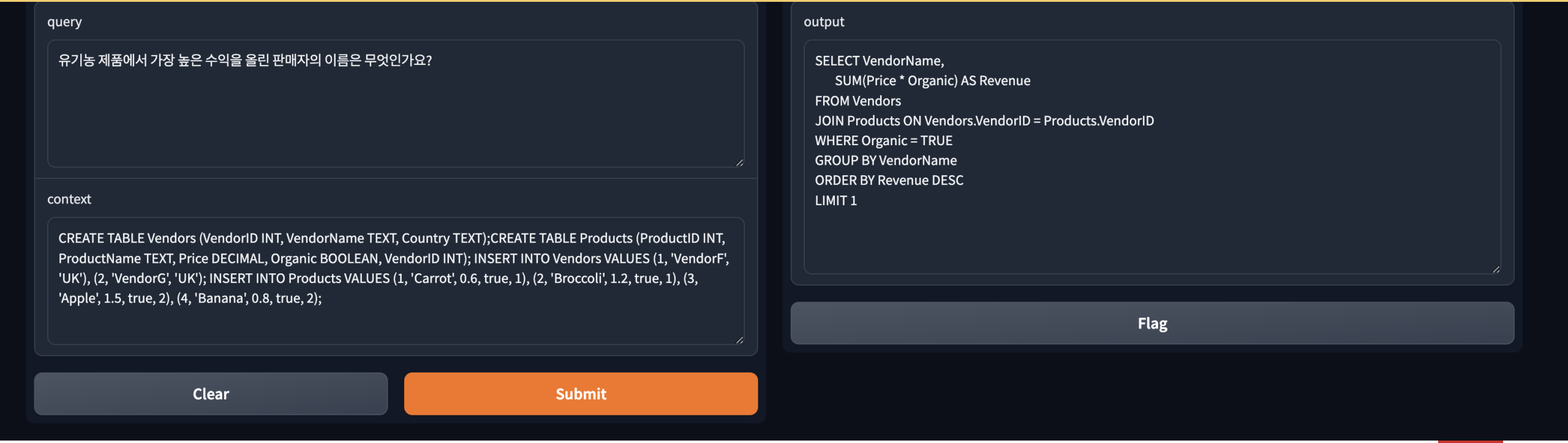
output

```
SELECT VendorName,
    SUM(Price * Organic) AS Revenue
FROM Vendors
JOIN Products ON Vendors.VendorID = Products.VendorID
WHERE Organic = TRUE
GROUP BY VendorName
ORDER BY Revenue DESC
LIMIT 1
```

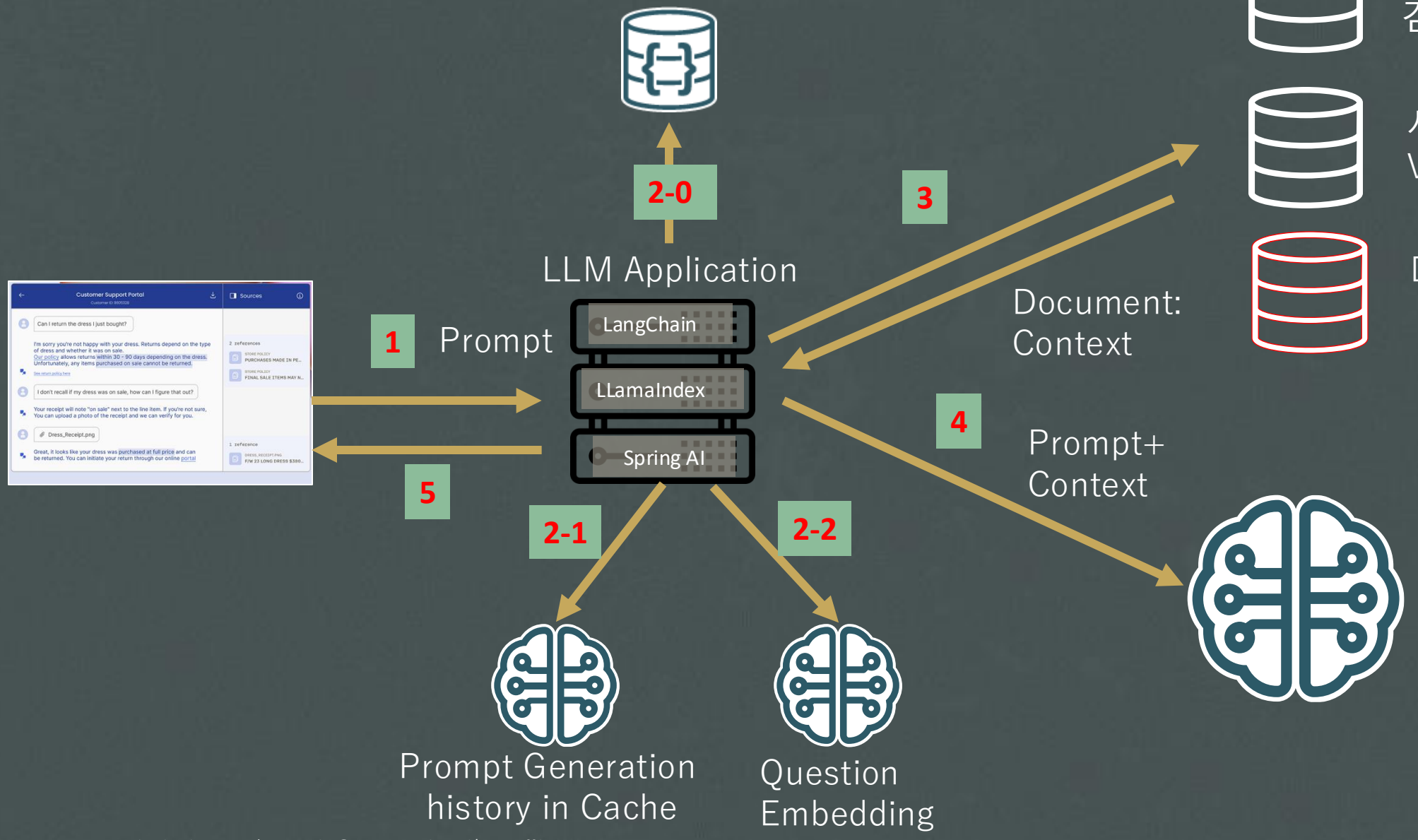
Flag

NL2SQL, Text to SQL ➔ N Step RAG

- 사용자 질의에서 관련 엔티티 추출
- 사용자 엔티티 메타 정보 제공
  - Schema Data: Table, Column, Comment, Table Partitioning….
  - Few Shot => Join 예시



RAG(Retrieval Augmented Generation) 아키텍처:



Oracle의 전략 3

LLM & Human을 위한 데이터 베이스

Oracle Database 23ai : NEW LTS focused on AI

Data Use Case Domains

Schema Privileges

ORACLE Database 23c

Readable PDB Standby

Operational Property Graphs

Real-time SQL Plan Management

Lock-Free Reservations

Transactional Microservices

JSON / Relational Duality

AI Vector Search

True Cache

SQL Firewall

Priority Transactions

Globally Distributed Database w/ RAFT

Rolling Patching

JavaScript Stored Procedures

Developer Role

Shrink Tablespace

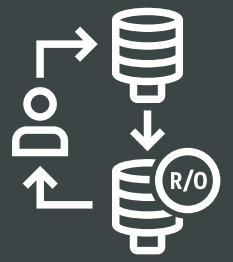
Cross-tier Precheck Constratins

Oracle Database 23ai : NEW LTS focused on AI

Data Use Case Domains

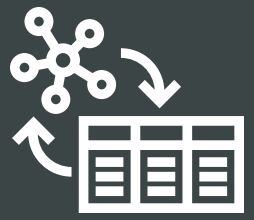
Schema Privileges

ORACLE Database 23ai



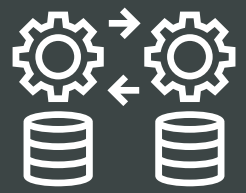
Readable PDB Standby

Operational Property Graphs



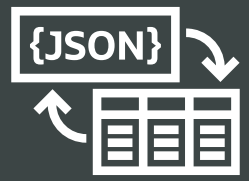
Real-time SQL Plan Management

Lock-Free Reservations



Transactional Microservices

JSON / Relational Duality



AI Vector Search

True Cache

SQL Firewall

Priority Transactions

Rolling Patching

JavaScript Stored Procedures

Developer Role

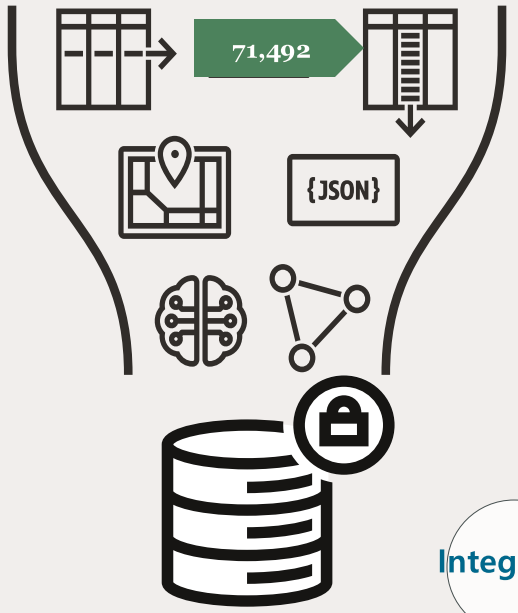
Shrink Tablespace

Cross-tier Precheck Constratins

Globally Distributed Database w/ RAFT

AI Vector Search 특징 For LLM
벡터 임베딩 부터 벡터 검색까지 End-to-End 관리 지원

New In 23ai

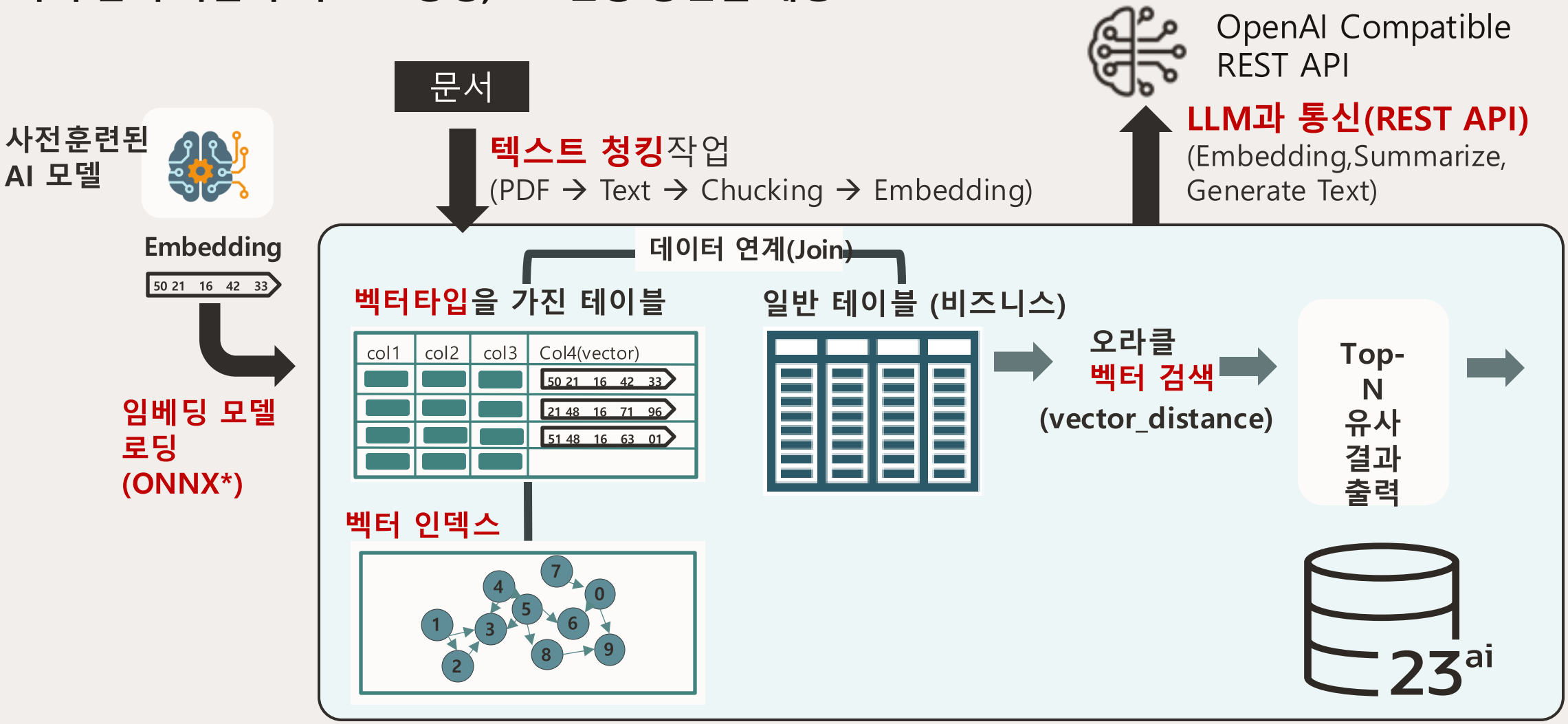


- **Generate** 비정형 데이터에서 임베딩 벡터 생성
- **Store** 테이블의 벡터 컬럼에 벡터 데이터 저장
- **Index** 저장되어 있는 벡터에 대한 인덱스 지원
- **Search** SQL로 벡터 검색을 수행
- **Integrate** 미션 크리티컬 엔터프라이즈 기능과의 통합

```
CREATE TABLE my_images (
   id number, metadata JSON,
   image BLOB,img_vec VECTOR);
```

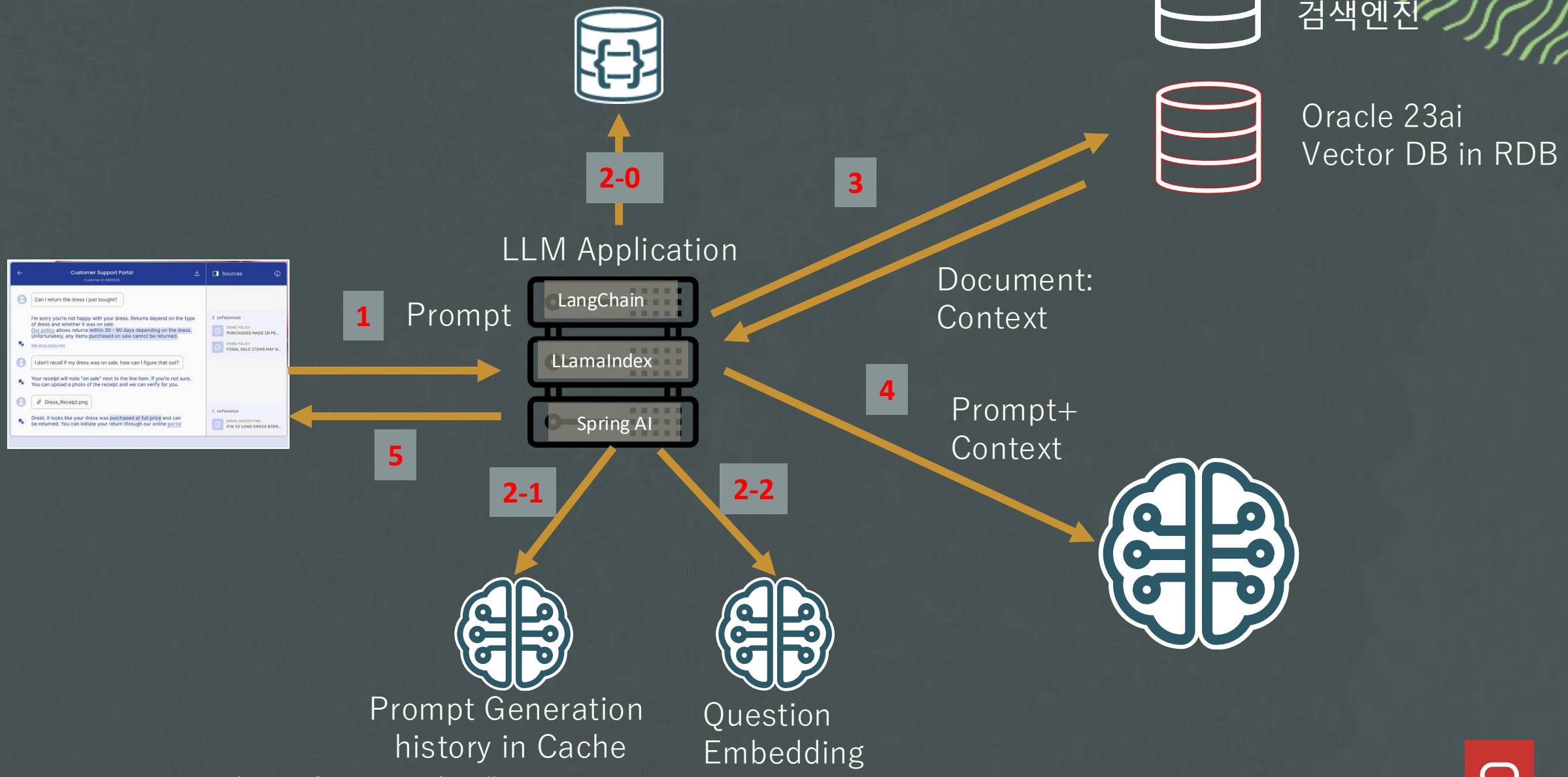
```
SELECT image, metadata
FROM my_images
ORDER BY VECTOR_DISTANCE(img_vec,
:query_vec)
FETCH FIRST 5 ROWS ONLY;
```

AI Vector Search 특징 For LLM
벡터 검색 기술과 텍스트 청킹, LLM연동 방안을 제공



*ONNX (Open Neural Network eXchange) : ML모델을 표현하는 오픈된 표준형식

RAG(Retrieval Augmented Generation) 아키텍처:

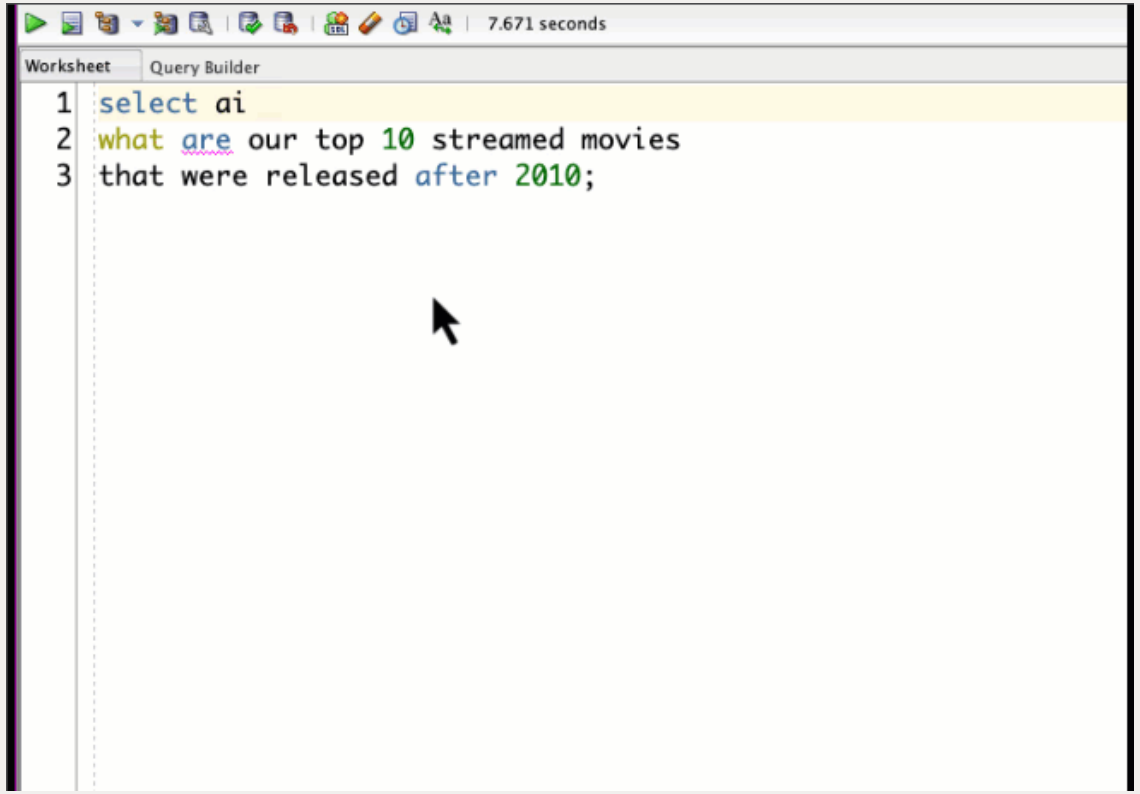


Select AI

자연어를 사용하여 SQL을 통해 데이터베이스 및 LLM과 상호 작용

- 직접 LLM과 자연어 기반으로 질문 응답
- RAG를 위해 스키마 메타데이터와 벡터 데이터베이스 활용
- 자연어로 SQL 쿼리 생성, 실행, 설명
- 다양한 AI 키워드 제공

  (chat, narrate, showsql, runsql, explainsql)



Deploy

Oracle Select AI

데모 순서

- 1-1. DB 접속 & Select AI 설정
- 1-2. Select AI 기초
- 1-3. SH 스키마 쿼리 데모
- 1-4. Movie 테이블 추가
- 1-5. Movie 프로파일 등록
- 1-6. Movie 데이터 조회
- 2-1. Select AI RAG
- 3-1. NoCode RAG: OCI Gen AI Agent
- 4-1. Naive RAG With ADB & Cohere

SH Query 목록

2000년도의 총 판매량은 얼마인가요

SH 복합 쿼리 실행

- RUNSQL: 2000년도의 총 판매량은 얼마인가요

```
select ai runsql 2000년도의 총 판매량은 얼마인가요
```

	Total_Sales_Quantity
0	232,646

- SHOWSQL: 2000년도의 총 판매량은 얼마인가요

```
select ai showsql 2000년도의 총 판매량은 얼마인가요
```

	RESPONSE
0	SELECT SUM("S"."QUANTITY_SOLD") AS "Total_Sales_Quantity" FROM "SH"."SALES"

```
SELECT SUM("S"."QUANTITY_SOLD") AS "Total_Sales_Quantity"
FROM "SH"."SALES" "S"
JOIN "SH"."TIMES" "T" ON "S"."TIME_ID" = "T"."TIME_ID"
WHERE "T"."CALENDAR_YEAR" = 2000
```

- Na

```
select ai narrate 2000년도의 총 판매량은 얼마인가요(한글로 출력해주세요)
```

	RESPONSE
0	2000년도의 총 판매량은 232,646입니다.

Deploy

Oracle Select AI

데모 순서

- 1-1. DB 접속 & Select AI 설정
- 1-2. Select AI 기초
- 1-3. SH 스키마 쿼리 데모
- 1-4. Movie 테이블 추가
- 1-5. Movie 프로파일 등록
- 1-6. Movie 데이터 조회
- 2-1. Select AI RAG
- 3-1. NoCode RAG: OCI Gen AI Agent
- 4-1. Naive RAG With ADB & Cohere

SH Query 목록

Nason Mann 이 구매한 상품들의 총액은 얼마인가요

SH 복합 쿼리 실행

- RUNSQL: Nason Mann 이 구매한 상품들의 총액은 얼마인가요

```
select ai runsql Nason Mann 이 구매한 상품들의 총액은 얼마인가요
```

	Total_Amount_Sold
0	67,773

- SHOWSQL: Nason Mann 이 구매한 상품들의 총액은 얼마인가요

```
select ai showsql Nason Mann 이 구매한 상품들의 총액은 얼마인가요
```

	RESPONSE
0	SELECT SUM("S"."AMOUNT_SOLD") AS "Total_Amount_Sold" FROM "SH"."SALES" "S" JOIN "SH"."CUSTOMERS" "C" ON "S"."CUST_ID" = "C"."CUST_ID" WHERE "C"."CUST_FIRST_NAME" = 'Nason' AND "C"."CUST_LAST_NAME" = 'Mann'

- Na

```
select ai narrate Nason Mann 이 구매한 상품들의 총액은 얼마인가요(한글로 출력해주세요)
```

	RESPONSE
0	Nason Mann이 구매한 상품들의 총액은 67,773입니다.

Oracle Select AI

데모 순서

- 1-1. DB 접속 & Select AI 설정
- 1-2. Select AI 기초
- 1-3. SH 스키마 쿼리 데모
- 1-4. Movie 테이블 추가
- 1-5. Movie 프로파일 등록
- 1-6. Movie 데이터 조회
- 2-1. Select AI RAG
- 3-1. NoCode RAG: OCI Gen AI Agent
- 4-1. Naive RAG With ADB & Cohere

- 설정 코드: Select AI 대상 프로파일

```
BEGIN
    DBMS_CLOUD_AI.CREATE_PROFILE(
        profile_name => 'OPENAI',
        attributes => '{
            "provider": "openai",
            "credential_name": "OPENAI_CRED",
            "object_list": [
              {"owner": "SH","name":"CHANNELS"},
              {"owner": "SH","name":"COSTS"},
              {"owner": "SH","name":"COUNTRIES"},
              {"owner": "SH","name":"CUSTOMERS"},
              {"owner": "SH","name":"PRODUCTS"},
              {"owner": "SH","name":"PROMOTIONS"},
              {"owner": "SH","name":"SALES"},
              {"owner": "SH","name":"TIMES"},
              {"owner": "ADMIN","name":"ACTOR"},
              {"owner": "ADMIN","name":"MOVIE"},
              {"owner": "ADMIN","name":"DIRECTOR"},
              {"owner": "ADMIN","name":"movie_actor"}
            ],
            "max_tokens":512,
            "stop_tokens": [";"],
            "model": "gpt-4o",
            "temperature": 0.1,
            "comments": true
        }'
    );
END;
```

Deploy

Oracle Select AI

데모 순서

- 1-1. DB 접속 & Select AI 설정
- 1-2. Select AI 기초
- 1-3. SH 스키마 쿼리 데모
- 1-4. Movie 테이블 추가
- 1-5. Movie 프로파일 등록
- 1-6. Movie 데이터 조회
- 2-1. Select AI RAG
- 3-1. NoCode RAG: OCI Gen AI Agent
- 4-1. Naive RAG With ADB & Cohere

자연어 쿼리

2020년 최다 관객 동원 배우 3명을 알려주세요

Movie 쿼리 실행

- RUNSQL: 2020년 최다 관객 동원 배우 3명을 알려주세요

```
select ai runsql 2020년 최다 관객 동원 배우 3명을 알려주세요
```

	Actor_Name	Total_Audience
0	이종석	10,000,000
1	박보검	3,800,000
2	김혜수	900,000

- SHOWSQL: 2020년 최다 관객 동원 배우 3명을 알려주세요

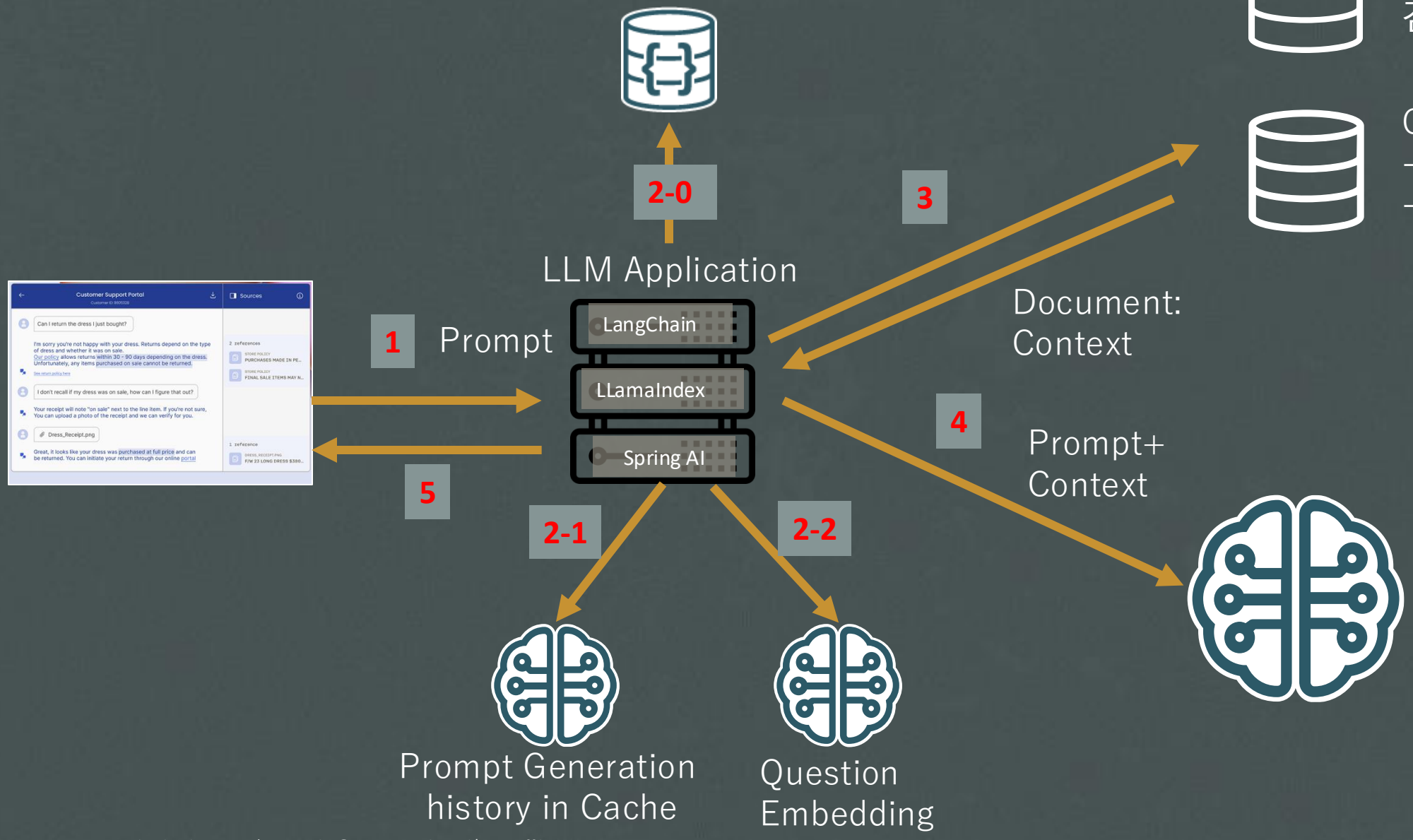
```
select ai showsql 2020년 최다 관객 동원 배우 3명을 알려주세요
```

	RESPONSE
0	SELECT "A"."NAME" AS "Actor_Name", SUM("M"."AUDIENCE_COUNT") AS "Total_Audience" FROM "ADMIN"."MOVIE_ACTOR" "MA" JOIN "ADMIN"."ACTOR" "A" ON "MA"."ACTOR_ID" = "A"."ACTOR_ID" JOIN "ADMIN"."MOVIE" "M" ON "MA"."MOVIE_ID" = "M"."MOVIE_ID" WHERE EXTRACT(YEAR FROM "M"."RELEASE_DATE") = 2020 GROUP BY "A"."NAME" ORDER BY "Total_Audience" DESC FETCH FIRST 3 ROWS ONLY

- Na

```
sele                                                세요)
```

RAG(Retrieval Augmented Generation) 아키텍처:



Ask Oracle E-Business Suite (EBS)

sai

Prompt

show the workorder with the most total scrap quantity.

Ask

Workorder Name	Workorder Id	Scrap Quantity	Assembly Name	Assembly Description	Assembly Category	Assembly Uom
G95-WO1	1011352	7005	G95-Mirror Assembly	G95-Mirror Assembly		Ea

1 - 1

Ask Oracle E-Business Suite (EBS)

sai

Prompt

Are there any quality issues reported for the assembly of this workorder

Ask

Collection Plan	Issue Description	Severity	Cost	Status	Date Opened	Date Closed	Nonconformance Number	Workorder	Assembly	Total Scrap Quantity
NTP1_DISP	In process nonconformance	MEDIUM		PENDING	2/14/2022		NC189	G95-WO1	G95-Mirror Assembly	7005
NTP1_DISP	Out of spec assembly	HIGH	1700	CLOSED	2/8/2022	2/8/2022	NC188	G95-WO1	G95-Mirror Assembly	7005
NTP1_DISP	In process damaged assembly	HIGH	1000	CLOSED	2/8/2022	2/8/2022	NC187	G95-WO1	G95-Mirror Assembly	7005
NTP1_NCM_MAST	Assembly issue	HIGH	4000	NEW	2/14/2022		NC202	G95-WO1	G95-Mirror Assembly	7005
NTP1_NCM_MAST	Out of spec assembly	HIGH	1700	CLOSED	2/8/2022	2/8/2022	NC188	G95-WO1	G95-Mirror Assembly	7005
NTP1_NCM_MAST	In process nonconformance	HIGH	2000	NEW	4/17/2023		NC195	G95-WO1	G95-Mirror Assembly	7005

Ask Oracle E-Business Suite (EBS)

sai

Prompt

are any of these issues due to a faulty component

Ask

Collection Plan	Issue Description	Severity	Cost	Status	Date Opened	Date Closed	Nonconformance Number	Workorder	Assembly	Total Scrap Quantity	Faulty Component
NTP1_DISP	Out of spec assembly	HIGH	1700	CLOSED	2/8/2022	2/8/2022	NC188	G95-WO1	G95-Mirror Assembly	7005	Plate Glass
NTP1_DISP	In process nonconformance	MEDIUM		PENDING	2/14/2022		NC189	G95-WO1	G95-Mirror Assembly	7005	Plate Glass
NTP1_DISP	In process damaged assembly	HIGH	1000	CLOSED	2/8/2022	2/8/2022	NC187	G95-WO1	G95-Mirror Assembly	7005	Plate Glass
NTP1_NCM_MAST	Assembly issue	HIGH	4000	NEW	2/14/2022		NC202	G95-WO1	G95-Mirror Assembly	7005	Plate Glass
NTP1_NCM_MAST	In process nonconformance	HIGH	2000	NEW	4/17/2023		NC195	G95-WO1	G95-Mirror Assembly	7005	Plate Glass

하이브리드 검색 데모
벡터 검색과 키워드 검색

토끼가 잠을 잔 이유

거북이가 승리한 이유

하이브리드 검색

```
SELECT JSON_SERIALIZE(
       DBMS_HYBRID_VECTOR.SEARCH(
       json('{
           "hybrid_index_name": "my_hybrid_idx",
           "search_text": "토끼가 잠을 잔 이유",
           "search_fusion": "INTERSECT",
           "return": {
              "topN": 5
           }
         }')) ) txt
    FROM dual
```

키워드 + 백터 검색

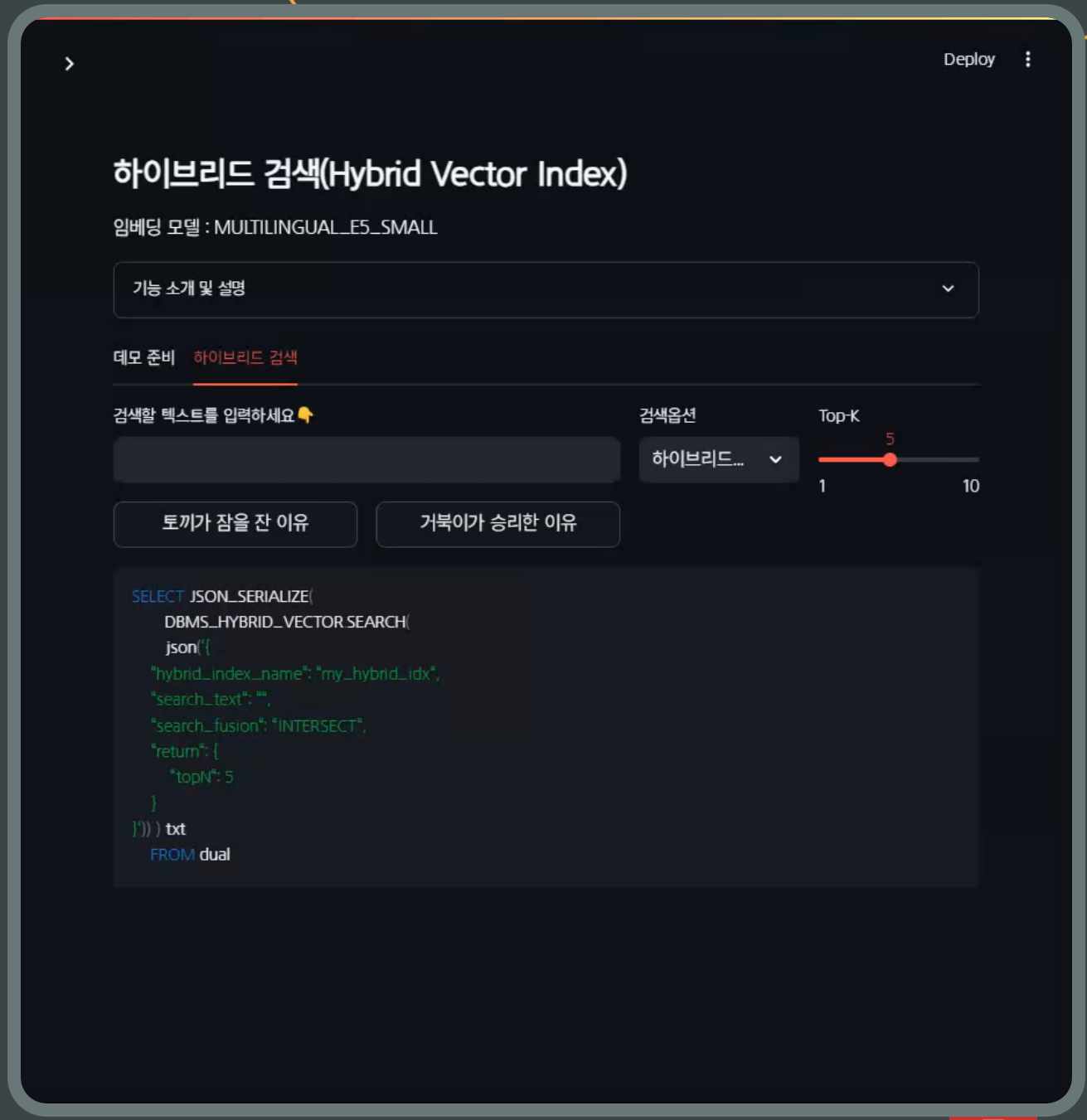
백터 검색

키워드검색

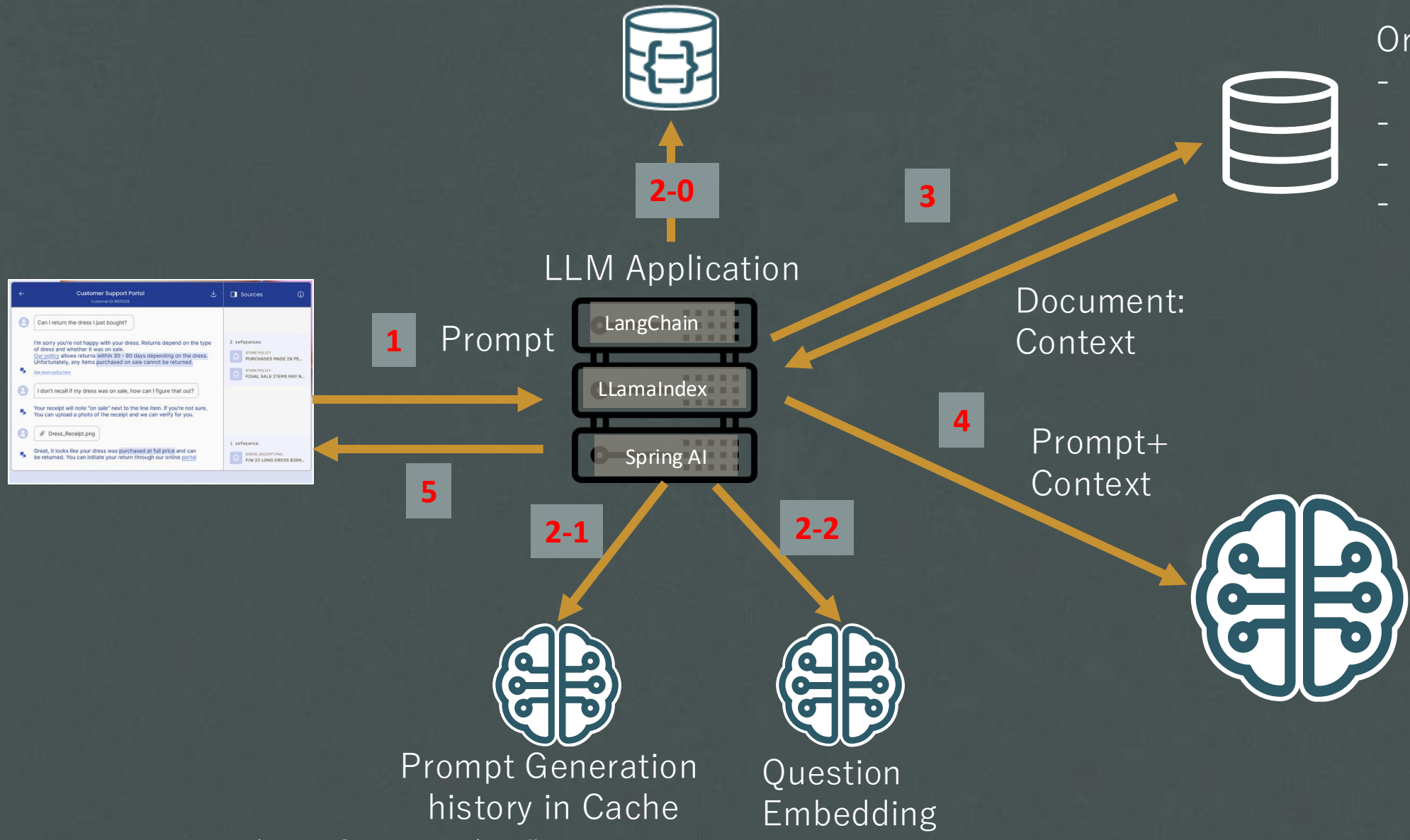
질문과 관련된 검색결과

✓ 한참 앞서간 토끼는 거북이가 자신을 따라잡지 못할 것이라 생각하고 길가에 누워 잠을 잤습니다

- 토끼는 웃으며 제안을 받아들였고, 경주가 시작되었습니다. (관계가 적은 데이터)
- 토끼는 자신의 빠른 다리를 자랑하며 친구들에게 늘 과시하곤 했습니다. (관계가 적은 데이터)



RAG(Retrieval Augmented Generation) 아키텍처:



AI Solutions Hub 바로 시작해 보세요

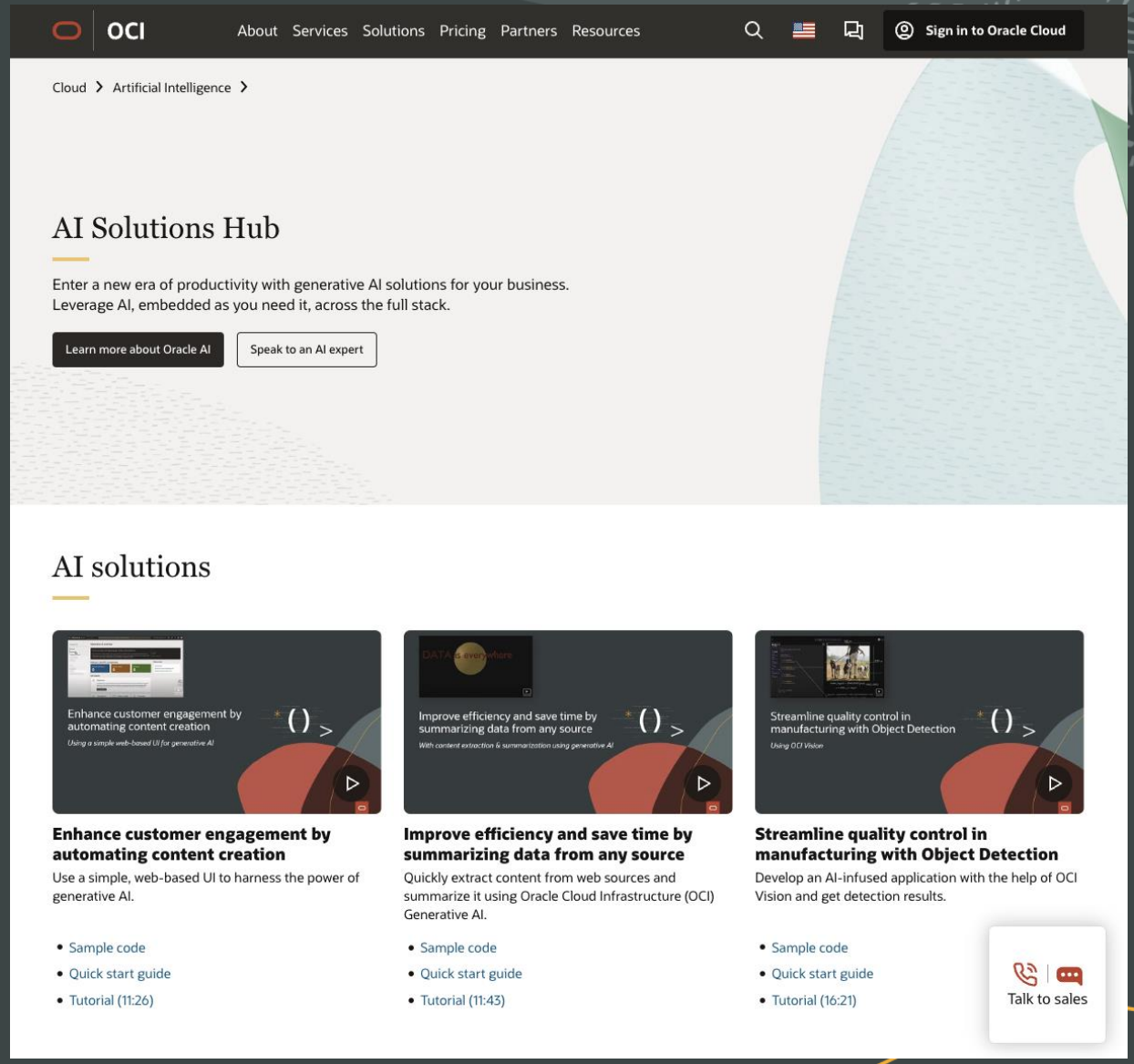
빠른 시작 AI 솔루션

자사 OCI 태넌시 솔루션 구성 가능

각 AI 솔루션 구성:

- 예제 코드
- 빠른 시작 가이드
- 튜토리얼 영상

새로운 솔루션은 계속 추가 됩니다



생성형 AI 시대에 오라클의 3개 전략

전략1: Oracle Enterprise LLM/Generative AI Platform

Model

DATA

Oracle Database 23C

ADW ATP EXACI EXACC

OpenSearch

Heatwave

INFRASTRUCTURE

H100/A100 /A10 GPU

Cloud Native OKE

Super Cluster

Chatbot

생성형 AI 시대에 오라클의 3개 전략

전략1: Oracle Enterprise LLM/Generative AI Platform

Model

DATA

Oracle Database 23C

ADW ATP EXACI EXACC

OpenSearch

Heatwave

INFRASTRUCTURE

H100/A100 /A10 GPU

Cloud Native OKE

Super Cluster

Chatbot

전략 2: 검증된 LLM

생성형 AI 시대에 오라클의 3개 전략

전략1: Oracle Enterprise LLM/Generative AI Platform

Model

DATA

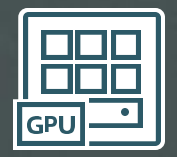
Oracle Database 23C

ADW ATP EXACI EXACC

OpenSearch

Heatwave

INFRASTRUCTURE



H100/A100 /A10 GPU

Cloud Native OKE

Super Cluster

Chatbot

전략 2: 검증된 LLM

전략 3: Gen AI를 위한 데이터 관리 체계: Oracle Database 23ai

생성형 AI 시대에 오라클의 3개 전략

전략1: Oracle Enterprise LLM/Generative AI Platform

Model

DATA

Oracle Database 23C

ADW ATP EXACI EXACC

OpenSearch

Heatwave

INFRASTRUCTURE

H100/A100 /A10 GPU

Cloud Native OKE

Super Cluster

Chatbot

전략 2: 검증된 LLM

전략 3: Gen AI를 위한 데이터 관리 체계: Oracle Database 23ai

감사합니다

김태완 상무

taewan.kim@oracle.com

Cloud Engineer Team

Oracle Korea