

---

AI 부트캠프 LLM 프로젝트

# LLM 프로젝트 OT

---

2025.10.28

박나연

# 박나연

---

현) 유튜브 <공원나연> 채널 운영 및 AI Agent 강사

현) AI Agent 엔지니어

전) 교육 AI 데이터사이언티스트

링크드인



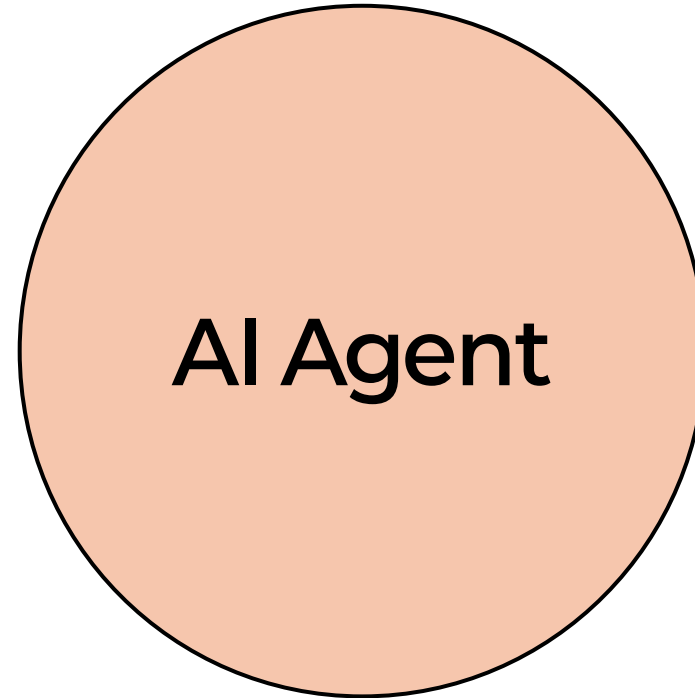
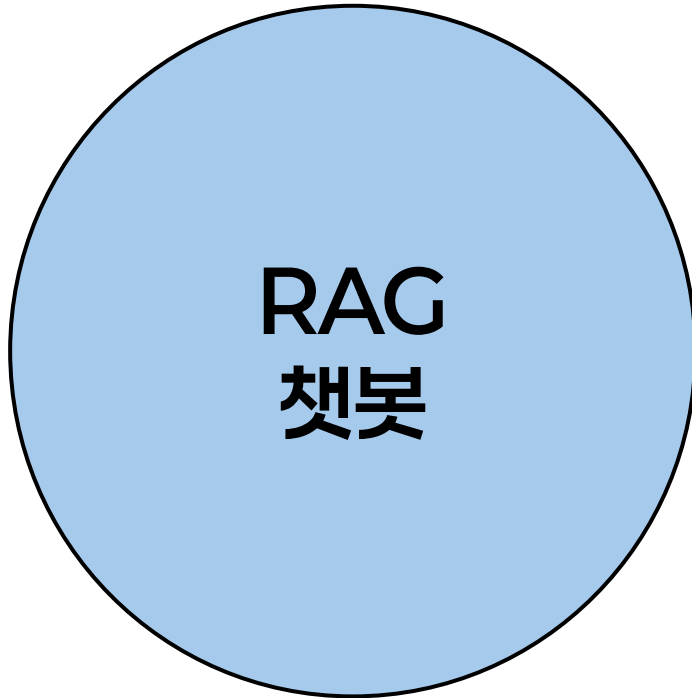
공원나연



# 멘토링 일정

- 10월 28일(화) 10:00~13:00(3H) - OT
- 10월 30일(목) 14:00~19:00(5H) - 멘토링
- 11월 04일(화) 14:00~19:00(5H) - 멘토링
- 11월 06일(목) 14:00~19:00(5H) - 발표회

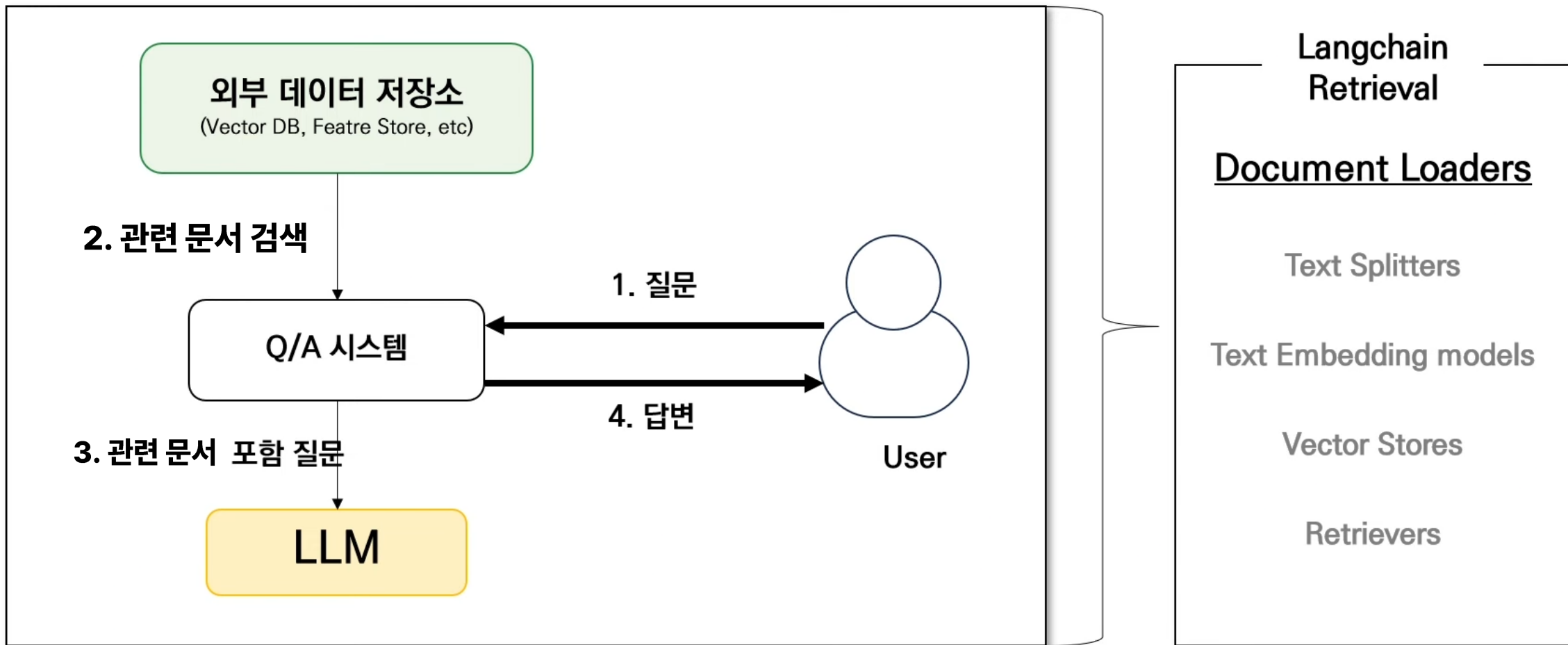
# 프로젝트 주제 소개



# RAG 챗봇 구조

수집한 문서 데이터를 지식베이스에 저장하고, 이를 기반으로 답변하는 질의응답 챗봇

## 〈RAG 구조〉



# 필수 기능

1. LLM의 답변을 의도대로 생성하기 위해 **System Prompt** 엔지니어링 (공통)
2. 필요한 데이터를 수집하는 기능
3. RAG를 하기 위해 데이터를 VectorDB에 저장하는 기능
4. Langchain를 통해 사용자 입력과 관련있는 데이터를 추출(검색조회)하는 기능
5. 멀티턴(Multi-turn) 기능 (공통)
6. 챗봇 UI 개발(Gradio, Streamlit 등 프레임워크 사용가능) (공통)
7. main.py를 통해서 나머지 모듈들을 불러와서 한번에 실행하는 기능 (챗봇의 답변 얻음) (공통)

# 선택 기능 (구현시 가산점)

1. 챗봇의 검색 혹은 답변 성능을 **평가**하는 기능
2. SQLite나 pymysql등을 사용하여 **RDBMS**에 구축된 데이터로부터 RAG를 수행하는 기능 (text-to-sql)
  - 여기서 사용하는 DB는 각자 프로젝트 기획에 맞는 table 데이터를 수집하여 사용
  - schema를 반드시 README.md에 기술

# Requirements

**1.개발 언어:** Python

**2.API 사용**

1. <https://platform.openai.com/docs/models>
2. <https://www.upstage.ai/pricing>

**3.모듈 구조:**

1. main.py
2. llm.py
3. prompts.py
4. retriever.py
5. vector\_database.py
6. demo.py
7. ...

**4. 오픈소스 라이브러리 사용 :** 제한 없음

**5. 코드 협업:** GitHub 사용, 모듈별 Branch, 모든 팀원이 최소한 1번 이상 PR 수행



# 참고 자료

랭체인 문서 전처리: <https://docs.langchain.com/oss/python/integrations/splitters>

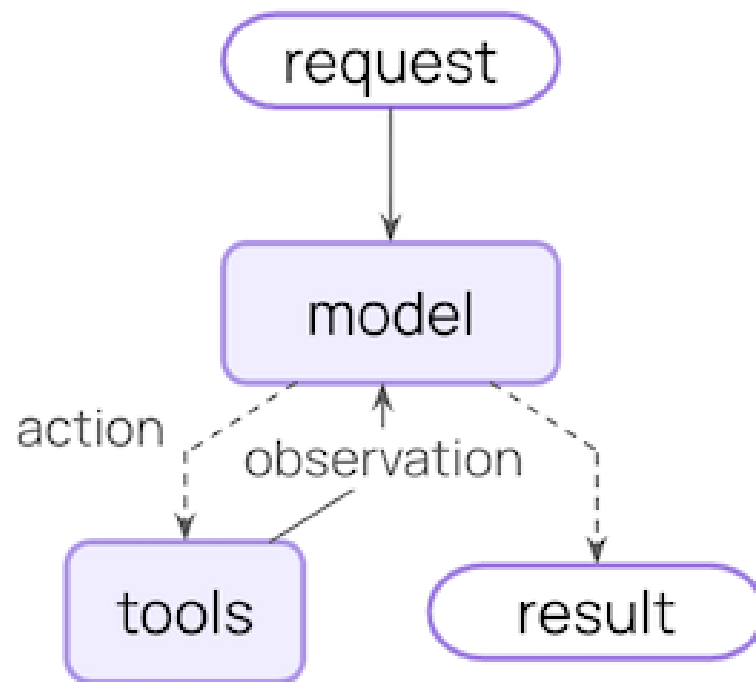
랭체인 크로마 DB: <https://python.langchain.com/docs/integrations/vectorstores/chroma/>

## 인터페이스 구축 프레임워크

- 그라디오: <https://www.gradio.app/>
- 스트림릿: <https://streamlit.io/>
- 오픈웹유아이: <https://openwebui.com/>

# AI Agent 구조

도구를 사용하여 퀄리티 좋은 답변을 생성하는 AI Agent 챗봇



# 필수 기능

1. LLM의 답변을 의도대로 생성하기 위해 **System Prompt** 엔지니어링 (공통)
2. 웹 검색 기능, 파일 저장 기능 필수 (도구)
3. 답변에 필요한 3개 이상의 도구를 사용(직접 구현한 도구가 포함되어야 함)
  - 특정 페르소나 필요 (ex. 여행 챗봇이면, 여행 관련 도구 생성 필수 - 여행계획 생성 도구, 구글 지도 검색 도구 등)
4. 멀티턴(Multi-turn) 기능 (공통)
5. 챗봇 UI 개발(Gradio, Streamlit 등 프레임워크 사용가능) (공통)
6. main.py를 통해서 나머지 모듈들을 불러와서 한번에 실행하는 기능 (챗봇의 답변 얻음) (공통)

# 선택 기능 (구현시 가산점)

1. 챗봇의 도구 호출 혹은 답변의 성능을 **평가**하는 기능
2. 성능 개선을 위한 노드, 엣지 추가(불필요하면 안됨)
3. 지식베이스(Vector DB, RDB 등)에 데이터를 수집해 적재하고, 이를 기반으로 답하는 RAG 기능

# Requirements

1. 개발 언어: Python

2. API 사용

1. <https://platform.openai.com/docs/models>

2. <https://www.upstage.ai/pricing>

3. 모듈 구조:

1. main.py

2. llm.py

3. prompts.py

4. tools.py

5. node.py (필요하면 노드 개수 만큼)

6. edge.py (선택)

7. make\_graph.py

8. demo.py

9. ...

4. 오픈소스 라이브러리 사용 : 제한 없음

5. 코드 협업: GitHub 사용, 모듈별 Branch, 모든 팀원이 최소한 1번 이상 PR 수행

# 참고 자료

ReAct 패턴 에이전트: <https://docs.langchain.com/oss/python/langchain/agents>

랭체인 도구 만드는 법: <https://docs.langchain.com/oss/python/langchain/tools#tools>

랭체인 도구 목록: <https://docs.langchain.com/oss/python/integrations/tools>

## 인터페이스 구축 프레임워크

- 그라디오: <https://www.gradio.app/>
- 스트림릿: <https://streamlit.io/>
- 오픈웹유아이: <https://openwebui.com/>

# 프로젝트 최종 제출

**발표 자료(PDF), README.md 포함 소스코드**

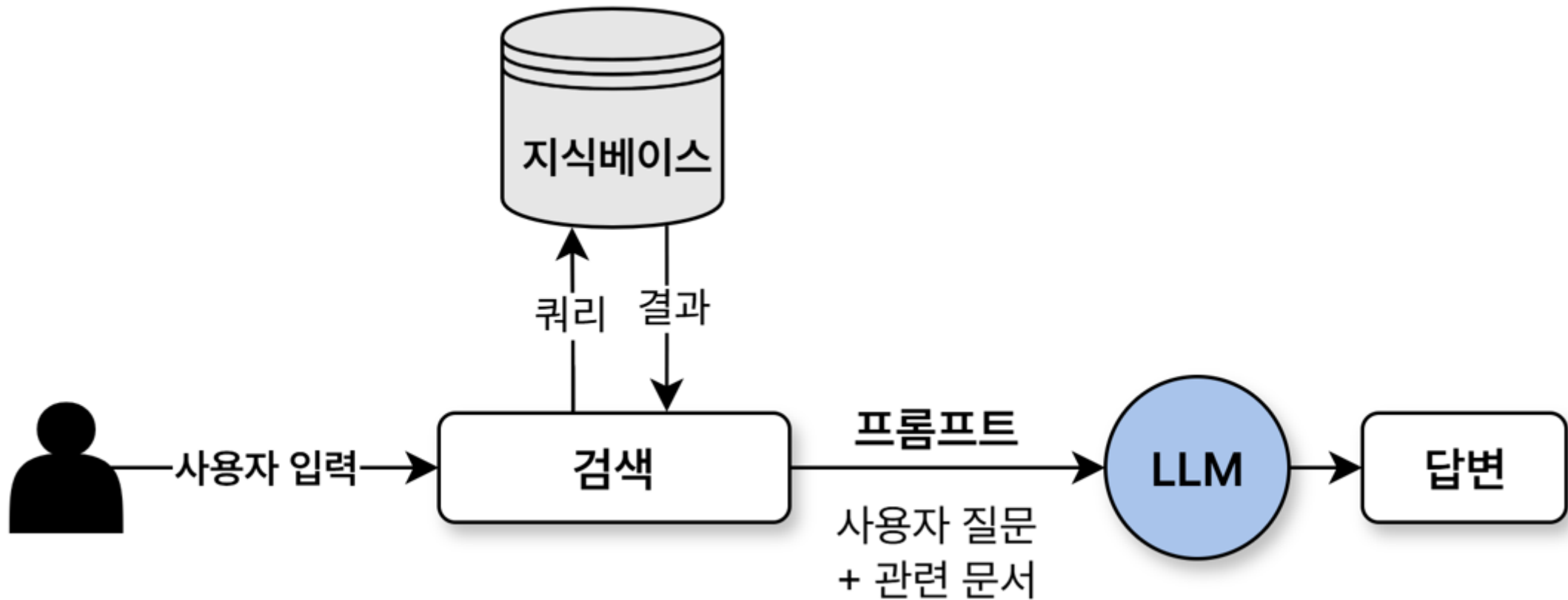
**발표 자료에 들어가야 할 것**

1. 어떤 문제 상황에 필요한 챗봇인지, 왜 이 챗봇을 만든건지 배경 설명
2. 챗봇에 활용될 수 있는 사용자 시나리오(10개) – 질문 리스트

# 프로젝트에 필요한 기술



# RAG란?



# RAG를 위한 데이터 적재

데이터 수집

데이터  
텍스트화

텍스트 분할

임베딩

DB 저장

<https://wikidocs.net/253706>

<https://docs.langchain.com/oss/python/integrations/splitters>

<https://console.upstage.ai/docs/capabilities/embed>  
<https://platform.openai.com/docs/guides/embeddings>

<https://python.langchain.com/docs/integrations/vectorstores/chroma/>

# Text2SQL 이란?

- 자연어 문장을 SQL 쿼리로 변환하는 기술

"올해 매출이 가장 높은 상품을 보여줘."

LLM

```
SELECT product_name  
FROM sales  
WHERE year = 2025  
ORDER BY revenue DESC  
LIMIT 1;
```

# Text2SQL 프롬프트

당신은 SQL 쿼리문을 생성하는 언어 모델입니다.

사용자로부터 질문이 주어지면, 주어진 데이터베이스 스키마를 기반으로 문법적으로 올바른 SQL 쿼리를 생성하세요.

질문에 특정한 개수의 예시를 요구하지 않는 이상, 항상 최대 5개 결과만 반환되도록 쿼리에 LIMIT을 포함하세요.

결과를 더 흥미롭게 만들기 위해 관련 있는 열을 기준으로 정렬해도 좋습니다.

INSERT, UPDATE, DELETE, DROP 등 DML 문은 절대 생성하지 마세요.

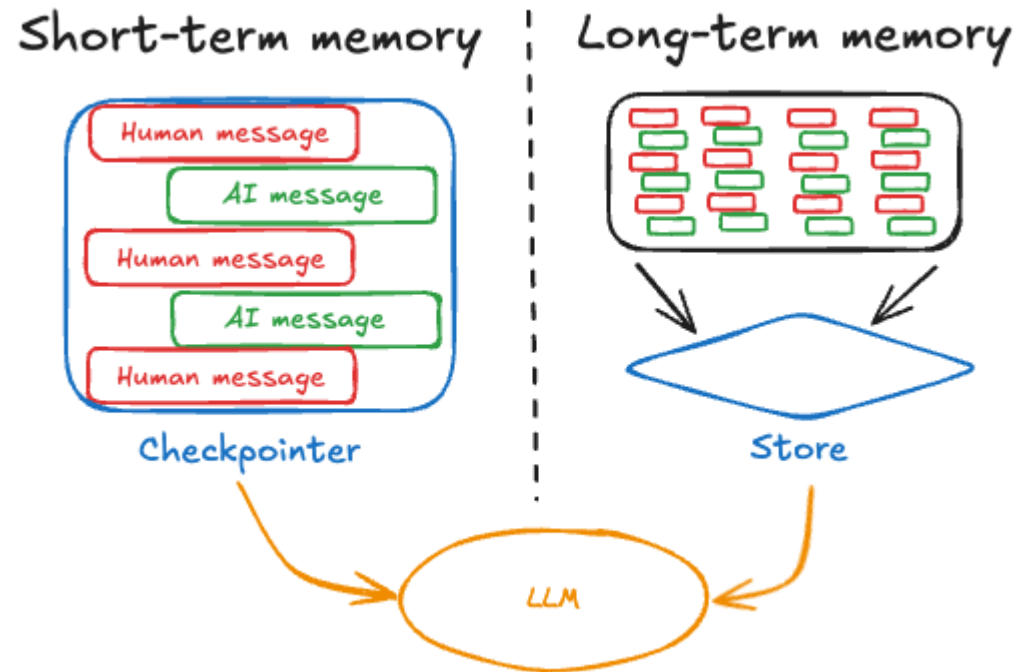
응답은 SQL 쿼리문만 출력하세요. 어떤 마크다운 기호(```sql)나 설명도 포함하지 마세요.

사용자 질문: {question}

데이터베이스 스키마: {schema}

생성된 SQL 쿼리:

# 멀티턴 구현 방법



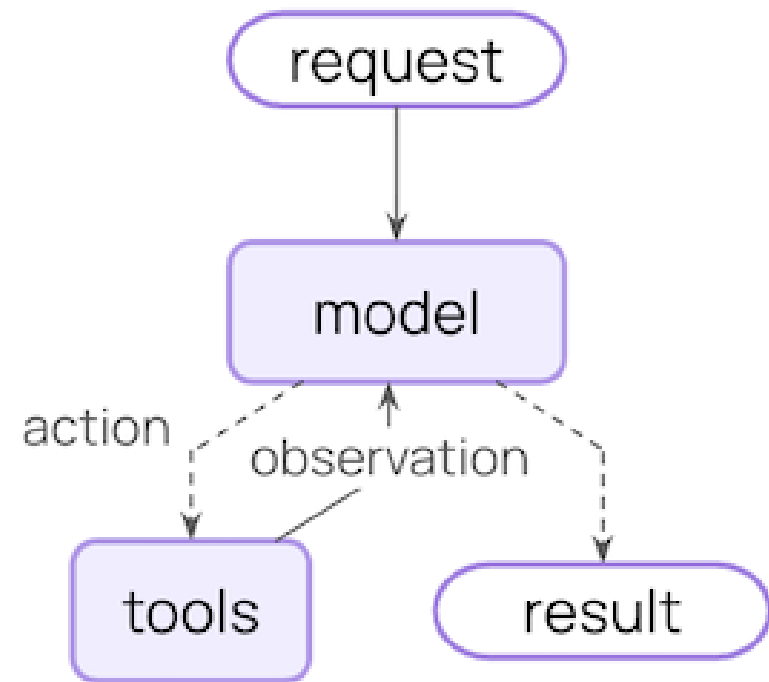
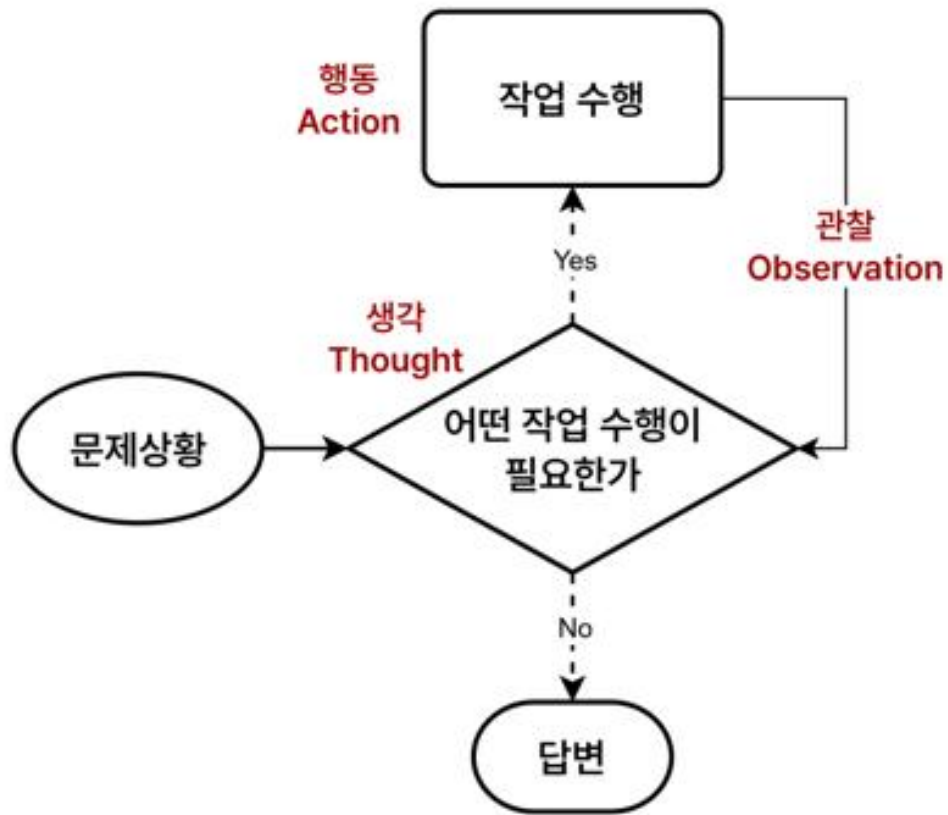
Langchain 사용시

: [https://python.langchain.com/docs/how\\_to/chatbots\\_memory/](https://python.langchain.com/docs/how_to/chatbots_memory/)

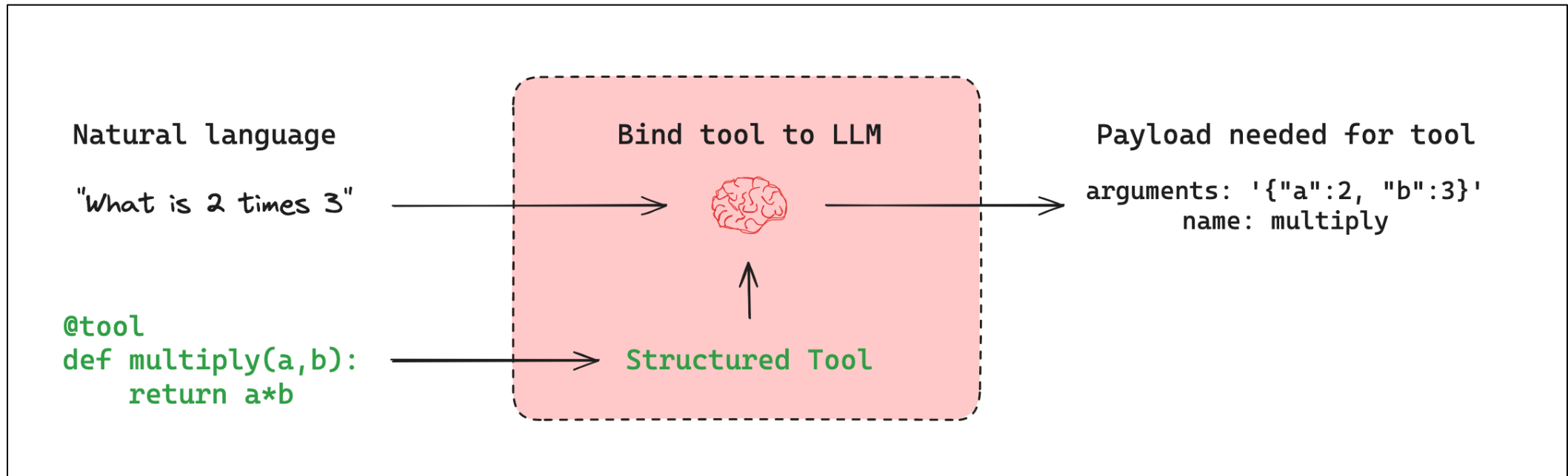
: <https://buly.kr/3COqTil>

Langgraph 사용시: <https://docs.langchain.com/oss/python/langgraph/add-memory>

# AI Agent란?



# 도구 호출이란?



사용자의 입력을 기반으로 도구 호출에 필요한 arguments를 감지하는 역할 수행

# [0813 특강 자료]

[웹검색 ReAct Agent 만들기]

[https://colab.research.google.com/drive/1n3vDwnESqk2LkxU15zAkRPwNu55ZTnTV?usp=drive\\_link](https://colab.research.google.com/drive/1n3vDwnESqk2LkxU15zAkRPwNu55ZTnTV?usp=drive_link)

[RAG Agent 만들기]

[https://colab.research.google.com/drive/1popw7xqUtbpZ10ZVcCPXcrU5WLXd-lv4?usp=drive\\_link](https://colab.research.google.com/drive/1popw7xqUtbpZ10ZVcCPXcrU5WLXd-lv4?usp=drive_link)



# **QnA 및 팀별 활동 진행**

**감사합니다 😊**