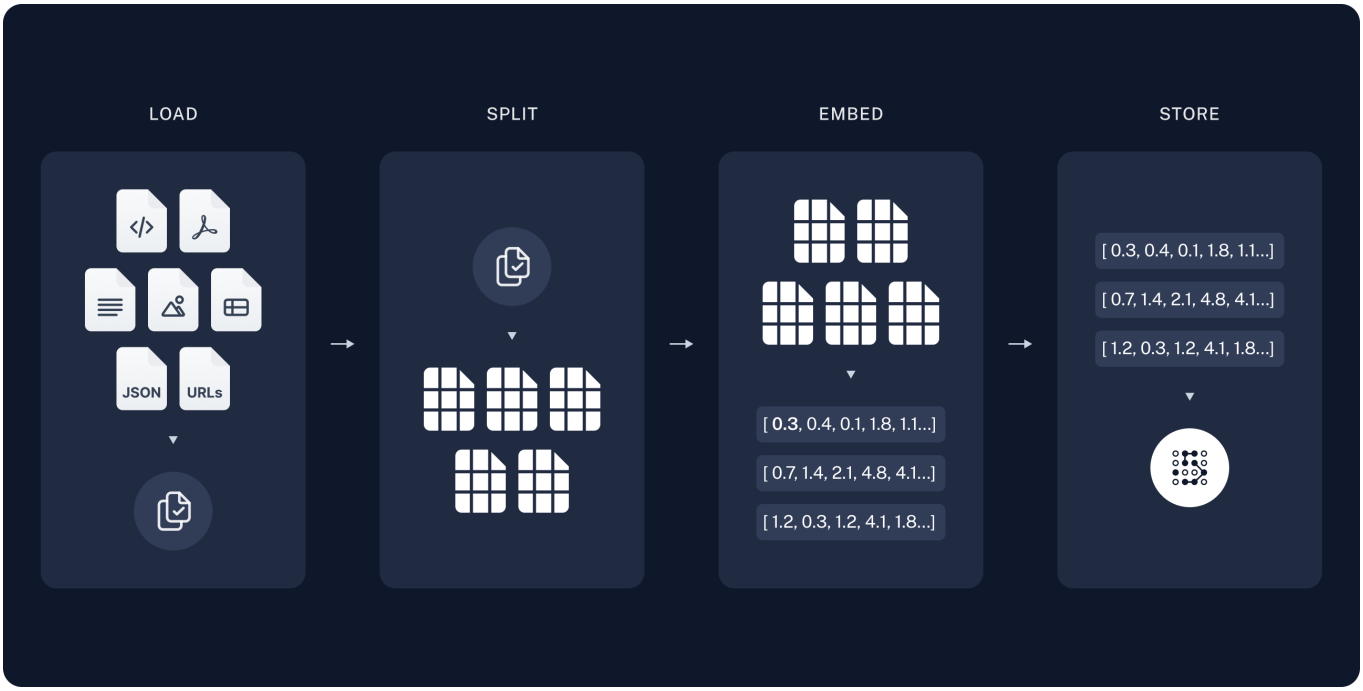
RAG

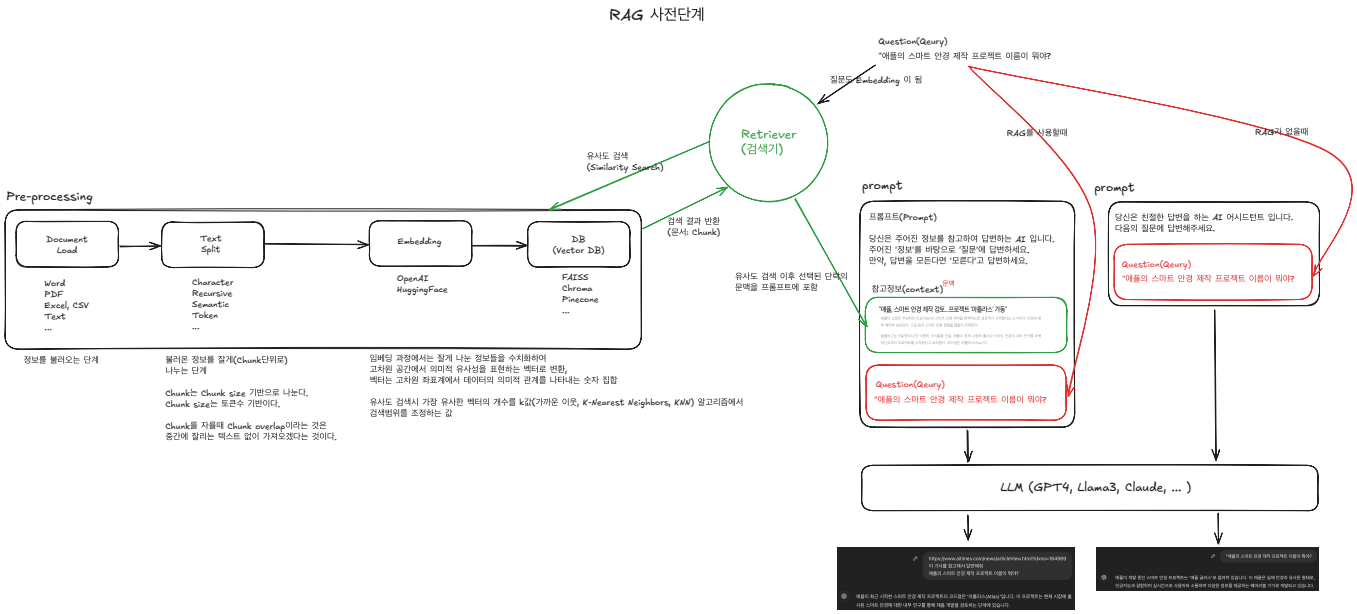
사전단계 (pre_processing)



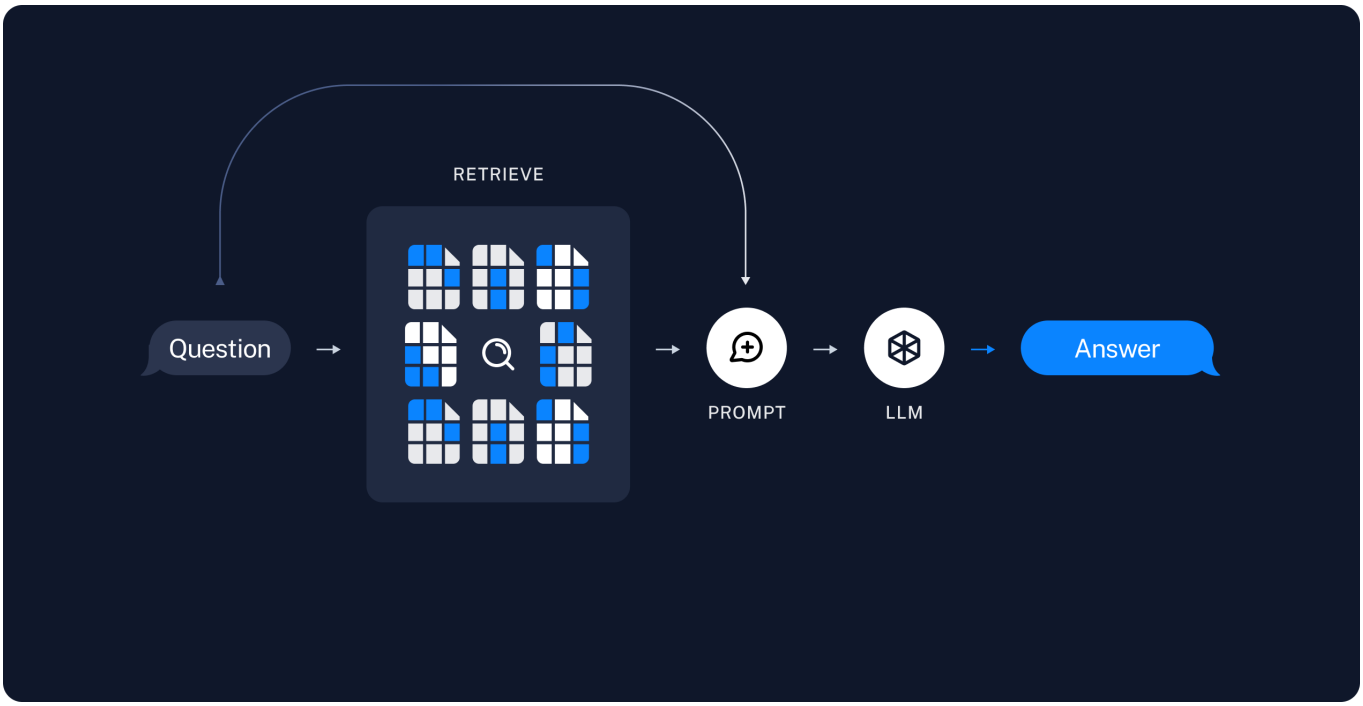
출처 : teddynote

사전 작업 단계에서는 데이터 소스를 Vector DB (저장소) 에 문서를 로드-분할-임베딩-저장 하는 4단계를 진행합니다.

- 1단계 문서로드(Document Load): 문서 내용을 불러옵니다.
- 2단계 분할(Text Split): 문서를 특정 기준(Chunk) 으로 분할합니다.
- 3단계 임베딩(Embedding): 분할된(Chunk) 를 임베딩하여 저장합니다.
- 4단계 벡터DB 저장: 임베딩된 Chunk 를 DB에 저장합니다.



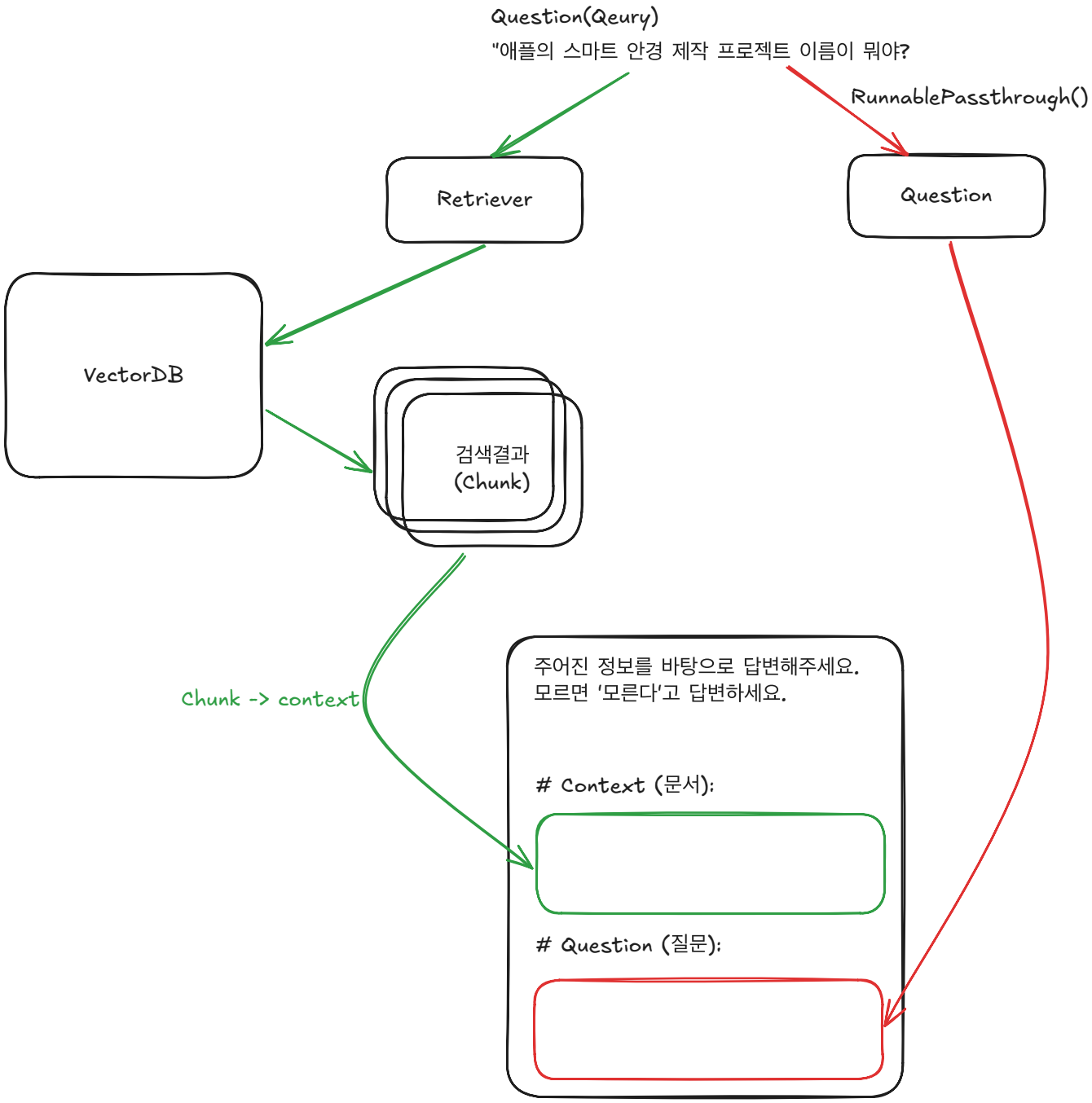
런타임 단계 (runtime_processing)



출처 : teddynote

- 5단계 검색기(Retriever): 쿼리(Query) 를 바탕으로 DB에서 검색하여 결과를 가져오기 위하여 리트리버를 정의합니다. 리트리버는 검색 알고리즘이며(Dense, Sparse) 리트리버로 나뉘게 됩니다. Dense: 유사도 기반 검색, Sparse: 키워드 기반 검색
- 6단계 프롬프트: RAG 를 수행하기 위한 프롬프트를 생성합니다. 프롬프트의 context 에는 문서에서 검색된 내용이 입력됩니다. 프롬프트 엔지니어링을 통하여 답변의 형식을 지정할 수 있습니다.
- 7단계 LLM: 모델을 정의합니다.(GPT-3.5, GPT-4, Claude, etc..)
- 8단계 Chain: 프롬프트 - LLM - 출력 에 이르는 체인을 생성합니다.

RAG 런타임 단계




```
# 체인 생성 예시입니다. 위의 그림을 참고해주세요.
chain = (
    {"context": retriever, "question": RunnablePassthrough()}
    | prompt
    | llm
```

```
    | StrOutputParser()
)
```