CHAPTER 06. SQL에 대화 저장

1. SQL (SQLAlchemy)

SQLAlchemy는 MIT 라이선스에 따라 배포되는 Python 프로그래밍 언어용 오픈 소스 SQL 툴킷이자 객체 관계 매퍼(ORM)입니다.

예제 – sql

```
from dotenv import load_dotenv
import os

load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')
```

```
storage를 사용하려면 다음 2가지만 제공하면 됩니다:

1. session_id : 사용자 이름, 이메일, 채팅 ID 등과 같은 세션의 고유 식별자입니다.
2. connection : 데이터베이스 연결을 지정하는 문자열입니다.
                이 문자열은 SQLAlchemy의 create_engine 함수에 전달됩니다.
```

```
from langchain_community.chat_message_histories import SQLChatMessageHistory
```

```
# SQLChatMessageHistory 객체를 생성하고 세션 ID와 데이터베이스 연결 파일을 설정
chat_message_history = SQLChatMessageHistory(
    session_id="sql_history",           # session_id의 기본 값으로 sql_history을 넣는다
    connection="sqlite:///sqlite.db"    # 데이터베이스 경로 지정 (sqlite.db는 파일 이름)
)
```

```
# 사용자 메시지를 추가합니다.
chat_message_history.add_user_message("안녕? 만나서 반가워. 내 이름은 이인환이야. 나는 파이썬 개발자야. 앞으로 잘 부탁해!")

# AI 메시지를 추가합니다.
chat_message_history.add_ai_message("안녕 이인환, 만나서 반가워. 나도 잘 부탁해!")
```

```
# 저장된 메시지 출력
print(chat_message_history.messages)
```

결과

```
[HumanMessage(content='안녕? 만나서 반가워. 내 이름은 이인환이야. 나는 파이썬 개발자야. 앞으로 잘 부탁해!'), AIMessage(content='안녕 이인환, 만나서 반가워. 나도 잘 부탁해!')]
```

2. SQL

SQLChatMessageHistory 클래스를 LCEL Runnables 와 쉽게 결합할 수 있습니다.

예제 – SQL LCEL Runnables

```
from dotenv import load_dotenv
import os

load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')
```

```
from langchain_core.prompts import (
    ChatPromptTemplate,
    MessagesPlaceholder,
)
from langchain_openai import ChatOpenAI
from langchain_community.chat_message_histories import SQLChatMessageHistory
from langchain_core.runnables.history import RunnableWithMessageHistory
from langchain_core.output_parsers import StrOutputParser
```

```
prompt = ChatPromptTemplate.from_messages(
    [
        ("system", "You are a helpful assistant."),             # 시스템 메시지
        MessagesPlaceholder(variable_name="chat_history"),      # 대화 기록을 위한 Placeholder
        ("human", "{question}"),                                # 질문
    ]
)
```

```
chain = prompt ¦ ChatOpenAI(model_name="gpt-4o-mini") ¦ StrOutputParser()
```

```
# sqlite.db에서 대화 내용을 가져오는 함수를 만듭니다.
def get_chat_history(user_id, conversation_id):
    return SQLChatMessageHistory(
        table_name=user_id,
        session_id=conversation_id,
        connection="sqlite:///sqlite.db",
    )
```

```
# config_fields 를 설정합니다. 이는 대화정보를 조회할 때 참고 정보로 활용합니다.
# user_id: 사용자 ID
# conversation_id: 대화 ID

from langchain_core.runnables.utils import ConfigurableFieldSpec

config_fields = [
    ConfigurableFieldSpec(
        id="user_id",                  #  get_chat_history 함수의 파라미터 이름과 같아야 한다.
        annotation=str,
        name="User ID",
        description="Unique identifier for a user.",
        default="",
        is_shared=True,
    ),

    ConfigurableFieldSpec(
        id="conversation_id",          #  get_chat_history 함수의 파라미터 이름과 같아야 한다.
        annotation=str,
        name="Conversation ID",
        description="Unique identifier for a conversation.",
        default="",
        is_shared=True,
    ),
]
```

```
chain_with_history = RunnableWithMessageHistory(
    chain,
    # 대화 기록을 가져오는 함수를 설정합니다.
    get_chat_history,

    # 입력 메시지의 키를 "question"으로 설정.프롬프트의 question와 맵핑.
    input_messages_key="question",

    # 대화 기록 메시지의 키를 "history"로 설정. MessagesPlaceholder의 chat_history 맵핑.
    history_messages_key="chat_history",

    # 대화 기록 조회시 참고할 파라미터를 설정합니다.
    history_factory_config=config_fields,
)
```

```
# "configurable" 키 아래에 "user_id", "conversation_id" key-value 쌍을 설정합니다.
config = {
    "configurable": {
        "user_id": "user1",
        "conversation_id": "conversation1"
    }
}
```

```
# 질문에 이름을 물어보는 질문을 해보겠습니다. 이전에 저장한 대화가 있다면, 올바르게 답할 것입니다.

# chain_with_history 객체의 invoke 메서드를 호출하여 질문에 대한 답변을 생성합니다.
# invoke 메서드에는 질문 딕셔너리와 config 설정이 전달됩니다.

# 질문과 config 를 전달하여 실행합니다.
chain_with_history.invoke({"question": "안녕 반가워, 내 이름은 이인환이야"}, config)
```

결과

```
'안녕하세요, 이인환님! 만나서 반갑습니다. 어떻게 도와드릴까요?'
```

```
chain_with_history.invoke({"question": "내 이름이 뭐라고?"}, config)
```

결과

```
'당신의 이름은 이인환이십니다! 맞나요?'
```

- 출처 : 랭체인LangChain 노트 (https://wikidocs.net/233343)