# 온 프레미스(On-Premise) 환경에서의 MCP(Model Context Protocol)를 활용한 범용 Text-to-SQL 에이전트 설계 및 구현에 관한 연구

김지섭[1], 김동우[1], 이재희[2]

[1]동서울대학교 컴퓨터 소프트웨어과 학부생

[2]동서울대학교 컴퓨터 소프트웨어과 교수

wltjq6702@gmail.com, gimdongu19@gmail.com, lih7314@du.ac.kr

# A Study on the Design and Implementation of an On-Premise General-Purpose Text-to-SQL Agent based on the MCP(Model Context Protocol)

Ji-Seob Kim[1], Dong-Woo Kim[1], Jae-Hee Lee[2]

[1] [2]Dept. of Computer Software, Dong-Seoul University

요 약

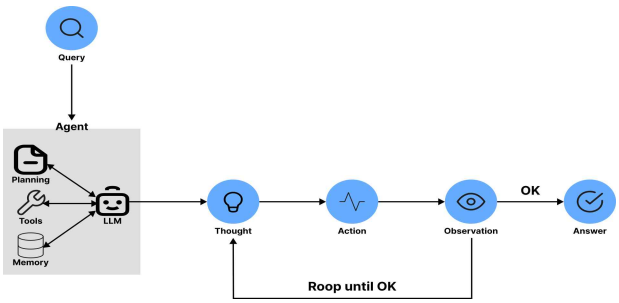
본 연구는 온 프레미스(On-Premise) 환경에서 MCP(Model Context Protocol)를 활용한 범용 Text-to-SQL 에이전트의 설계 및 구현에 관한 것이다. 본 연구에서 제안한 시스템은 직접 구현한 Host-Client-Server 구조로 동작하며 Streamlit 기반의 MCP Host를 통해 LangGraph 기반 LLM-based 에이전트가 MCP Server에 구현된 9개의 표준화된 도구(Tool)를 사용하여 데이터베이스와 상호작용하는 구조이다. 총 4종류의 데이터베이스와 연동실험을 통해 자연어 질의를 성공적으로 처리하는 실험결과를 통해 본 연구의 범용성과 실용성을 검증하였으며, 이를 통해 온 프레미스 환경에서도 안전하고 확장 가능한 범용 Text-to-SQL 에이전트의 구현 가능성을 제시하였다.

## Ⅰ. 서 론

비전문가에게 SQL은 데이터 접근의 큰 장벽이다. 이를 해결하기 위한 기존 Text-to-SQL 기술은 특정 데이터베이스에 종속되어 확장성에 한계가 있다. 최근 LLM(Large Language Model)을 활용한 방식은 성능이 뛰어나지만, 고비용 상용 API 의존과 민감 데이터의 외부 전송에 따른 보안 문제를 안고 있다[1]. 본 연구에서는 이러한 한계점들을 극복하기 위해 온 프레미스 환경으로 비용 및 보안 문제를 해결하고 MCP(Model Context Protocol)로 확장성 문제를 해결한 범용 Text-to-SQL 에이전트(Agent)를 제안하였다. 본 연구에서 제안하는 시스템은 Streamlit 기반의 MCP Host, LangGraph 프레임워크를 활용한 LLM-based 에이전트 기반의 MCP Client, 그리고 직접 개발한 9개의 도구(Tool)를 갖춘 MCP Server로 구성된다. 에이전트가 자체적으로 연결된 데이터베이스 종류를 파악하여 자연어로 전달된 질의를 각 데이터베이스 종류에 맞는 SQL언어로 변환하여 각 데이터베이스에 전달하여 안전하게 상호작용하도록 설계 및 구현하였다.

## Ⅱ. LLM-based 에이전트

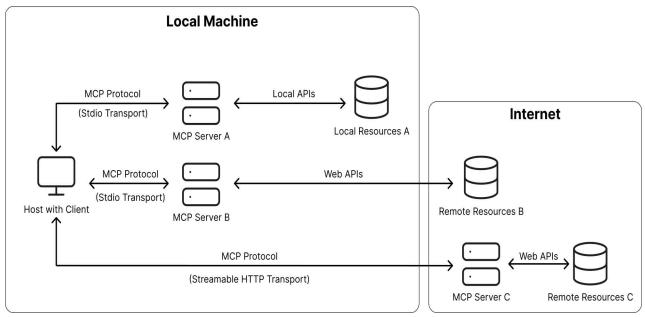
본 연구에서 제안하는 LLM-based 에이전트는 LangGraph의 React(Reasoning and Acting) 프레임워크를 기반으로 추론(Chain of Thought)과 행동(외부 도구 호출)을 교차 생성하도록 유도하였다[2]. 에이전트는 자연어 질의에 대해 생각(Thought)-행동(Action)-관찰(Observation) 단계를 반복하면서 질의를 해결한다. LLM이 추론 순위를 지정하여 행동 계획을 업데이트 하고, 행동을 통해 API 호출이나 데이터베이스 조회와 같은 외부 정보에 접근함으로써 추가 정보를 얻을 수 있다. 그림 1은 본 연구에서 제안한 LLM-based 에이전트의 구조이다.




[그림 1] LLM(Large Language Model)-based 에이전트의 구조

## Ⅲ. MCP (Model Context Protocol)

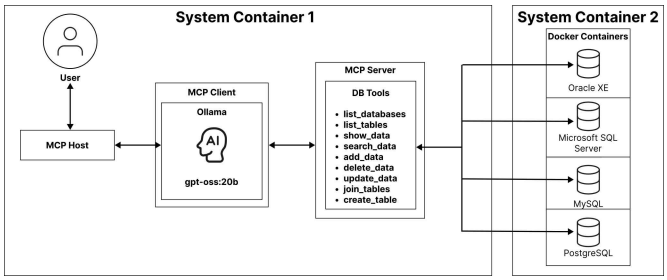
MCP는 2024년 앤트로픽(Anthropic)사에서 제안한 Host-Client-Server 구조로 동작하면서 LLM에 외부 도구와 데이터를 표준화된 방식으로 제공하는 개방형 프로토콜이다[3]. 이는 기존 통합 방식의 확장성 문제를 해결하고 개방적인 에이전트 구축을 지원한다. 그림 2는 본 연구에서 활용하는 MCP의 구조이다.




[그림 2] MCP(Model Context Protocol) 구조도

## Ⅳ. 제안한 시스템 구조도

제안한 시스템은 다음과 같이 직접 구현한 Host-Client-Server 구조로 동작한다. 사용자의 요청은 Streamlit 기반의 MCP Host를 통해 OpenAI사에서 공개한 gpt-oss:20b 모델을 활용한 LLM-based 에이전트 기반의 MCP Client에게 전달되며 MCP Client는 MCP Server의 9개의 도구를 호출하여 연결된 4개의 데이터베이스와 상호작용 후 최종 결과를 반환하는 구조이다. 본 연구에서는 도커(Docker)를 이용하여 에이전트 부분과 데이터베이스 부분을 2개의 컨테이너(Container)로 각각 분리하여 구현하였다. 그림 3은 본 연구에서 제안하는 시스템의 전체 구조이다.




[그림 3] 제안한 시스템 구조도

## Ⅴ. 실험 환경 및 내용

본 연구의 실험 환경 및 내용은 표 1과 같다.

<표 1> 실험 환경 및 내용

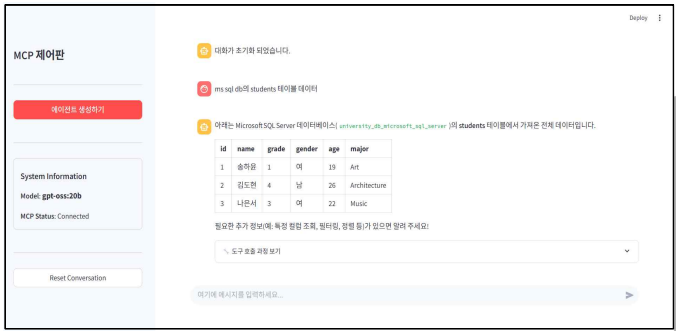
| 구분 | 항목 | 내용 |
|---|---|---|
| H/W | GPU | NVIDIA V100 32GB |
| S/W | OS | Debian 12.0 Bookworm |
| | Language | Python 3.12 |
| | LLM | gpt-oss:20b |
| | MCP Host | streamlit==1.44.1 |
| | MCP Client | langchain-ollama==0.3.6 |
| | | langgraph==0.3.21 |
| | MCP Server | mcp[cli]==1.6.0 |
| | | langchain-mcp-adapters ==0.0.7 |
| | Database | Oracle XE(21c), Microsoft SQL Server(2022), MySQL(8.0), PostgreSQL(15) |
| | Container | Docker 28.3.2 |

## Ⅵ. 제안한 시스템의 실험 결과

제안하는 Text-to-SQL 에이전트의 동작성능을 검증하기 위해 Oracle, Microsoft SQL Server, MySQL, PostgreSQL 등 4개의 데이터베이스 환경에서 에이전트의 Text-to-SQL 동작을 테스트했으며, 그림 4는 Oracle XE DB에 대한 테이블 목록 질의 결과이다.


[그림 4] Oracle XE DB에 테이블 목록 질의 결과

그림 5는 Microsoft SQL Server DB에 대한 테이블 데이터 질의 결과이다.

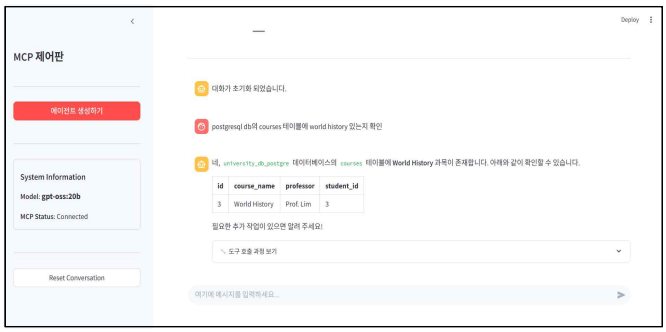



[그림 5] Microsoft SQL Server DB에 테이블 데이터 질의 결과

그림 6은 MySQL DB에 대한 데이터 추가 질의 결과이다.


[그림 6] MySQL DB에 데이터 추가 질의 결과

그림 7은 PostgreSQL DB에 대한 데이터 검색 질의 결과이다.




[그림 7] PostgreSQL DB에 데이터 검색 질의 결과

## Ⅶ. 결론

본 논문은 온 프레미스 환경에서 MCP를 활용해 비용, 보안, 확장성 문제들을 해결하기 위해 범용 Text-to-SQL 에이전트를 설계 및 구현하였다. 본 연구를 통해 온 프레미스 환경에서 충분히 실용적인 Text-to-SQL 에이전트의 가능성을 확인했다. 다만 현재의 Text-to-SQL 에이전트 구조는 복잡한 작업을 병렬적으로 처리하는 능력이 부족하다. 향후 A2A(Agent-to-Agent) 아키텍처 기반의 멀티 에이전트 구현에 대한 연구를 진행하고자 한다.

## 참 고 문 헌

[1] D. Gao, et al., "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation," arXiv preprint arXiv:2308.15363, 2023.

[2] S. Yao, et al., "ReAct: Synergizing Reasoning and Acting in Language Models," arXiv preprint arXiv:2210.03629, 2022.

[3] Xinyi Hou et al., "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions", arXiv:2503.23278, 2024.