**🧠 Using LangChain + Local Phi-3 model to query a SQL Server database**

**✅ No OpenAI.  
✅ Works with phi-3 model from Hugging Face.  
✅ Converts natural language questions into SQL and returns answers.  
✅ Runs locally on Windows with CPU.**

**🧱 What You’ll Set Up**

* **✅ A local LLM (Phi-3-mini)**
* **✅ LangChain’s SQL agent**
* **✅ SQL Server connection via pyodbc**
* **✅ Full working Python script**

**🔧 Step-by-Step Setup**

**✅ Step 1: Install dependencies**

**bash**

**CopyEdit**

**pip install langchain langchain-community transformers accelerate torch**

**pip install pyodbc sqlalchemy**

**✅ Step 2: Download the Phi-3 Model**

**Use the phi-3-mini-4k-instruct from Hugging Face:**

**python**

**CopyEdit**

**from transformers import AutoModelForCausalLM, AutoTokenizer, pipeline**

**from langchain.llms import HuggingFacePipeline**

**model\_id = "microsoft/phi-3-mini-4k-instruct"**

**tokenizer = AutoTokenizer.from\_pretrained(model\_id, trust\_remote\_code=True)**

**model = AutoModelForCausalLM.from\_pretrained(**

**model\_id,**

**device\_map="auto", # Use CPU/GPU automatically**

**trust\_remote\_code=True**

**)**

**pipe = pipeline(**

**"text-generation",**

**model=model,**

**tokenizer=tokenizer,**

**max\_new\_tokens=256,**

**temperature=0.2,**

**do\_sample=True,**

**)**

**llm = HuggingFacePipeline(pipeline=pipe)**

**✅ Step 3: Connect to SQL Server**

**Make sure you have ODBC Driver 17 for SQL Server installed.**

**python**

**CopyEdit**

**import urllib**

**from langchain\_community.utilities import SQLDatabase**

**params = urllib.parse.quote\_plus(**

**"DRIVER={ODBC Driver 17 for SQL Server};"**

**"SERVER=localhost;"**

**"DATABASE=mydb;"**

**"UID=sa;"**

**"PWD=yourPassword123"**

**)**

**connection\_uri = f"mssql+pyodbc:///?odbc\_connect={params}"**

**db = SQLDatabase.from\_uri(connection\_uri)**

**✅ Step 4: Create SQL Agent with LangChain**

**python**

**CopyEdit**

**from langchain\_experimental.sql import SQLDatabaseChain**

**db\_chain = SQLDatabaseChain.from\_llm(llm, db, verbose=True)**

**✅ Step 5: Ask Natural Language Questions**

**python**

**CopyEdit**

**question = "What was the total expense for Project Alpha in Q1?"**

**response = db\_chain.run(question)**

**print("\n💡 Answer:", response)**

**✅ Full Python Script Template**

**python**

**CopyEdit**

**# phi3\_sql\_agent.py**

**from transformers import AutoModelForCausalLM, AutoTokenizer, pipeline**

**from langchain.llms import HuggingFacePipeline**

**from langchain\_community.utilities import SQLDatabase**

**from langchain\_experimental.sql import SQLDatabaseChain**

**import urllib**

**# 1. Load Phi-3 model locally**

**print("🔄 Loading Phi-3...")**

**model\_id = "microsoft/phi-3-mini-4k-instruct"**

**tokenizer = AutoTokenizer.from\_pretrained(model\_id, trust\_remote\_code=True)**

**model = AutoModelForCausalLM.from\_pretrained(model\_id, device\_map="auto", trust\_remote\_code=True)**

**pipe = pipeline(**

**"text-generation",**

**model=model,**

**tokenizer=tokenizer,**

**max\_new\_tokens=256,**

**temperature=0.2,**

**do\_sample=True,**

**)**

**llm = HuggingFacePipeline(pipeline=pipe)**

**# 2. Connect to SQL Server**

**print("🔌 Connecting to SQL Server...")**

**params = urllib.parse.quote\_plus(**

**"DRIVER={ODBC Driver 17 for SQL Server};"**

**"SERVER=localhost;"**

**"DATABASE=mydb;"**

**"UID=sa;"**

**"PWD=yourPassword123"**

**)**

**connection\_uri = f"mssql+pyodbc:///?odbc\_connect={params}"**

**db = SQLDatabase.from\_uri(connection\_uri)**

**# 3. Create the SQL agent**

**db\_chain = SQLDatabaseChain.from\_llm(llm, db, verbose=True)**

**# 4. Run query**

**question = "What was the total expense for Project Alpha in Q1?"**

**print(f"\n❓ Question: {question}")**

**response = db\_chain.run(question)**

**print("\n💡 Answer:", response)**

**⚠️ Notes & Tips**

* **🔐 NEVER expose your DB password in production code.**
* **🧠 Tune prompt templates if SQL generation fails — Phi-3 is small but very capable with structured prompting.**
* **🚀 For better performance, try quantized Phi-3 or run with GPU.**

**Would you like me to help:**

* **Add prompt templates to improve SQL accuracy?**
* **Convert this into a web UI (like Gradio or Streamlit)?**
* **Log and validate generated SQL before executing?**

**Let me know how you'd like to expand it!**

**Ask ChatGPT**