

Construindo um SQL Agent Inteligente

Objetivo

Criar um **SQL Agent inteligente** que receba perguntas em linguagem natural (NLP), converta-as automaticamente em **queries SQL** e execute-as em um banco de dados PostgreSQL. O sistema deve interpretar perguntas complexas e retornar respostas formatadas, lidando com consultas simples e avançadas (JOINS, filtros e agregações).

Além disso, o desafio inclui diferenciais técnicos para mostrar sua capacidade de construção de soluções escaláveis e modernas com múltiplos agentes.

Requisitos Essenciais

- Criar um banco de dados PostgreSQL com tabelas inter-relacionadas.
 - Implementar um SQL Agent com **LangChain** para converter linguagem natural em SQL.
 - Criar um fluxo com **LangGraph** para processar a consulta.
 - Executar a query gerada e retornar os resultados formatados.
 - Garantir segurança contra **SQL Injection** e aplicar otimizações nas queries.
-

Estrutura do Banco de Dados

Trabalhe com três tabelas interligadas:

- clientes: Informações dos clientes.
- transacoes: Compras realizadas por cada cliente.
- produtos: Produtos disponíveis para venda.

Relacionamentos:

- Um cliente pode ter várias transações (1:N).
 - Cada transação está associada a um único produto (N:1).
-

Exemplos de Perguntas

- Quais clientes compraram um Notebook?
 - Quanto cada cliente gastou no total?
 - Quem tem saldo suficiente para comprar um Smartphone?
-

Fluxo de Execução

1. Usuário faz uma pergunta em linguagem natural.
 2. LangChain processa e converte a entrada em uma query SQL.
 3. A query é validada, otimizada e executada no PostgreSQL.
 4. Os resultados são retornados ao usuário, formatados de maneira compreensível.
-

Diferenciais / Adicionais Esperados

Para se destacar, implemente também:

- **RAG (Retrieval-Augmented Generation):** utilize o conteúdo do banco ou documentação como fonte adicional para gerar melhores queries.
 - **Arquitetura MCP (Model Context Protocol):** padronize o contexto compartilhado entre agentes para garantir coerência e escalabilidade.
 - **Memória Persistente Multisessão:** permita que os agentes mantenham o contexto de conversas passadas por usuário.
 - **Orquestração Multi-Agente com Papéis Diferentes:** implemente múltiplos agentes especializados (ex: parser NLP, validador SQL, executor, formatador de resposta).
 - **Observabilidade e Tracing:** registre logs detalhados de interações, geração de query, execuções e eventuais erros, com visualização de métricas (ex: via OpenTelemetry).
-

Critérios de Avaliação

- ✓ Queries SQL corretas e seguras
 - ✓ Interpretação precisa das perguntas
 - ✓ Código modular e bem estruturado
 - ✓ Implementação dos diferenciais será um forte destaque
-

Entrega

Crie um repositório público no GitHub com a solução e compartilhe o link com a recrutadora ao final.