**Como integrar OpenAI GPT-3.5 Turbo para geração de SQL**

**1. Criar conta na OpenAI**

* Acesse <https://platform.openai.com/signup>
* Cadastre-se com e-mail e cartão internacional (pode ser pré-pago).
* Após login, acesse **API Keys** no menu lateral e clique em **Create new secret key**.
* Copie e salve a chave gerada (aparece só uma vez).

**2. Adicionar a chave no seu app**

* **No Streamlit:**
  + Crie o arquivo .streamlit/secrets.toml (caso ainda não exista)
  + Adicione:

OPENAI\_API\_KEY = "sua\_chave\_aqui"

* **No Python puro:**
  + Guarde a chave em variável de ambiente ou arquivo de configuração seguro.

**3. Exemplo de código Python para usar GPT-3.5 Turbo**

import requests

def gerar\_sql\_com\_openai(pergunta, estrutura, api\_key):

url = "https://api.openai.com/v1/chat/completions"

headers = {

"Authorization": f"Bearer {api\_key}",

"Content-Type": "application/json"

}

system\_prompt = (

"Você é um assistente de BI. Gere sempre apenas a query SQL solicitada, "

"usando exclusivamente as tabelas e colunas listadas abaixo. "

"Nunca explique, nunca justifique, nunca escreva nada além do SQL. "

"Estrutura do banco:\n"

f"{estrutura}"

)

data = {

"model": "gpt-3.5-turbo",

"messages": [

{"role": "system", "content": system\_prompt},

{"role": "user", "content": pergunta}

],

"max\_tokens": 400,

"temperature": 0.1

}

resp = requests.post(url, headers=headers, json=data, timeout=60)

resp.raise\_for\_status()

resposta = resp.json()["choices"][0]["message"]["content"]

return resposta.strip()

# Exemplo de uso

estrutura = "orders(order\_id, order\_date, product\_name, quantity, total\_price)"

pergunta = "Qual o produto mais vendido em março de 2025?"

api\_key = "SUA\_API\_KEY\_AQUI"

sql\_gerado = gerar\_sql\_com\_openai(pergunta, estrutura, api\_key)

print(sql\_gerado)

**4. Recomendações**

* Faça perguntas em português, mas teste também em inglês se quiser comparar.
* Revise sempre o SQL antes de executar.
* Monitore os gastos pelo painel da OpenAI.

**5. Observações**

* Sem mensalidade: paga só o que usar.
* Controle total de gastos pelo painel.
* Fácil integração com qualquer aplicação Python.