

NLP SUNU

- Nlp dersi kapsamında bir proje yapacağım dedim. Fikir önerilerini de LLM'den aldım. (Gemini 3.0)
- Önerileri doğrultusunda şu öneriyi seçtim :
 - Yapılması planlanan proje :
 - **Öneri 5: "Doğal Dilden SQL Sorgusu Oluşturucu (Text-to-SQL)"**
 - **Amaç:** Yazılım bilmeyen bir yöneticinin "Geçen ay en çok satış yapan 5 ürünü listele" yazarak veri tabanından sonuç almasını sağlamak.
 - **Kullanılacak Tool'lar:** Python, OpenAI API, SQLite (Örnek veritabanı).
 - **Süreç:**
 1. Veritabanı şeması LLM'e tanıtılır (System prompt).
 2. Kullanıcı Türkçe istekte bulunur.
 3. LLM bunu SQL koduna çevirir (SELECT * FROM sales...).
 4. Sistem kodu çalıştırır ve sonucu tablo olarak gösterir.
 - **Sonuç (Demo):** Veri tabanına Türkçe soru sorup doğru verinin geldiğinin gösterilmesi.

Sonrasında gemini 3.0 a projeyi direkt olarak geliştir dedim. Kodları oluşturdu.

Hazırlık Aşaması

Öncelikle bilgisayarında Python'un kurulu olması gerekiyor. Proje klasöründe terminali açıp gerekli kütüphaneleri yükleyelim:

<> Bash

```
pip install openai streamlit pandas
```

- **openai:** Yapay zeka ile konuşmak için.
- **streamlit:** Hızlıca web arayüzü (demo) yapmak için.
- **pandas:** Veritabanından gelen veriyi düzgün bir tablo olarak göstermek için.

Adım 1: Örnek Veritabanını Oluşturma

Önce elimizde sorgulayabileceğimiz bir veri olması lazım. Bunun için küçük bir Python scripti ile `sqlite` veritabanı oluşturalım.

Dosya adı: `db_olustur.py`

<> Python

```
import sqlite3

def create_database():
    conn = sqlite3.connect('magaza.db')
    c = conn.cursor()

    # 1. Tablo: Ürünler
    c.execute('''
        CREATE TABLE IF NOT EXISTS urunler (
            id INTEGER PRIMARY KEY,
            ad TEXT,
            kategori TEXT,
            fiyat REAL
        )
    ''')

    # 2. Tablo: Satışlar
    c.execute('''
        CREATE TABLE IF NOT EXISTS satislar (
            id INTEGER PRIMARY KEY,
            urun_id INTEGER,
            miktar REAL,
            tarih TEXT
        )
    ''')
```

```
import sqlite3
```

```
def create_database():
```

```
    conn = sqlite3.connect('magaza.db')
```

```
    c = conn.cursor()
```

```
    # 1. Tablo: Ürünler
```

```
    c.execute('''
```

```
        CREATE TABLE IF NOT EXISTS urunler (
```

```
            id INTEGER PRIMARY KEY,
```

```
        ad TEXT,  
        kategori TEXT,  
        fiyat REAL  
    )  
''')
```

2. Tablo: Satışlar

```
c.execute('''  
    CREATE TABLE IF NOT EXISTS satislar (  
        id INTEGER PRIMARY KEY,  
        urun_id INTEGER,  
        adet INTEGER,  
        tarih DATE,  
        FOREIGN KEY (urun_id) REFERENCES urunler (id)  
    )  
''')
```

Örnek Veriler Ekleyelim

Ürünler: Laptop, Telefon, Kulaklık vb.

```
urun_data = [  
    (1, 'iPhone 15', 'Elektronik', 50000),  
    (2, 'MacBook Air', 'Bilgisayar', 45000),  
    (3, 'AirPods', 'Aksesuar', 5000),  
    (4, 'Samsung S24', 'Elektronik', 48000),  
    (5, 'Logitech Mouse', 'Aksesuar', 1500)  
]
```

```
c.executemany('INSERT OR IGNORE INTO urunler VALUES (?, ?, ?, ?)', urun_data)
```

Satışlar: Geçen ay ve bu ay yapılan satışlar

```
satis_data = [  
    (1, 1, 2, '2023-10-05'), # 2 tane iPhone  
    (2, 2, 1, '2023-10-10'), # 1 tane MacBook  
    (3, 3, 5, '2023-10-12'), # 5 tane AirPods  
    (4, 1, 1, '2023-11-01'),  
    (5, 5, 10, '2023-10-15'), # 10 tane Mouse
```

```

        (6, 4, 3, '2023-10-20')
    ]
    c.executemany('INSERT OR IGNORE INTO satislar VALUES (?,?,?,?)', satis_data)

    conn.commit()
    conn.close()
    print("Veritabanı (magaza.db) başarıyla oluşturuldu!")

if __name__ == "__main__":
    create_database()

```

Adım 2: Ana Uygulama (Text-to-SQL)

Şimdi asıl projeyi yapıyoruz. Kullanıcıdan metin alacağız, OpenAI'a "Bunu SQL'e çevir" diyeceğiz ve sonucu göstereceğiz.

Önemli: OpenAI API anahtarına ihtiyacın olacak. Kodun içindeki "BURAYA_API_KEY_GELECEK" kısmına kendi key'ini yazmalısın.

```

import streamlit as st
import sqlite3
import pandas as pd
from openai import OpenAI

# 1. Ayarlar ve API Key
# Gerçek projelerde API Key'i environment variable olarak saklamak daha güvenlidir.
client = OpenAI(api_key="BURAYA_SK_ILE_BASLAYAN_API_KEYINIZI_YAZIN")

# 2. Veritabanı Şemasını Tanımla (System Prompt için)
# LLM'in doğru SQL yazabilmesi için tablo yapısını bilmesi gerekir.
db_schema = """
Tablo: urunler
Sütunlar: id (INTEGER), ad (TEXT), kategori (TEXT), fiyat (REAL)

```

Tablo: satılar

Sütunlar: id (INTEGER), urun_id (INTEGER), adet (INTEGER), tarih (DATE)

İlişki: satılar.urun_id, urunler.id ile eşleşir.

"""

```
def get_sql_from_llm(user_question):
```

```
    """Kullanıcı sorusunu SQL sorgusuna çevirir."""
```

```
    system_prompt = f"""
```

```
    Sen uzman bir veri analistisin ve SQLite uzmanısın.
```

```
    Aşağıdaki veritabanı şemasına göre, kullanıcının Türkçe sorusunu SQL sorgusuna çevir.
```

```
    ŞEMA:
```

```
    {db_schema}
```

```
    KURALLAR:
```

1. Sadece SQL kodunu döndür. Açıklama yapma, markdown kullanma.
2. SQL sorgusu SQLite ile uyumlu olmalı.
3. JOIN işlemleri gerekirse yap.

```
    """
```

```
response = client.chat.completions.create(
```

```
    model="gpt-3.5-turbo", # veya gpt-4
```

```
    messages=[
```

```
        {"role": "system", "content": system_prompt},
```

```
        {"role": "user", "content": user_question}
```

```
    ],
```

```
    temperature=0 # Yaratıcılığı kısip doğruluğu artırmak için 0 yapıyoruz
```

```
)
```

```
sql_query = response.choices[0].message.content.strip()
```

```
# Bazen LLM kodu ``sql ... `` içine alır, temizleyelim:
```

```
sql_query = sql_query.replace("`sql", "").replace("`", "").strip()
```

```
return sql_query
```

```

def run_query(sql_query):
    """SQL sorgusunu veritabanında çalıştırır."""
    try:
        conn = sqlite3.connect('magaza.db')
        df = pd.read_sql_query(sql_query, conn)
        conn.close()
        return df, None
    except Exception as e:
        return None, str(e)

# --- ARAYÜZ (Streamlit) ---
st.title("🤖 Doğal Dilden SQL Oluşturucu")
st.markdown("Yazılım bilmeden veritabanına soru sorun!")

# Kullanıcı Girişi
question = st.text_input("Sorusunu buraya yaz:", placeholder="Örn: En pahalı ürün hangisi?")

if st.button("Sorgula"):
    if question:
        with st.spinner("Yapay zeka SQL kodunu yazıyor..."):
            # 1. LLM'den SQL al
            generated_sql = get_sql_from_llm(question)

            st.subheader("Oluşturulan SQL Kodu:")
            st.code(generated_sql, language='sql')

            # 2. SQL'i çalıştır
            result_df, error = run_query(generated_sql)

            if error:
                st.error(f"Hata oluştu: {error}")
            else:
                st.subheader("Sonuç:")
                if not result_df.empty:

```

```

        st.dataframe(result_df)
    else:
        st.warning("Bu sorgu sonucunda veri bulunamadı.")
else:
    st.warning("Lütfen bir soru yazın.")

# Veritabanı Önizleme (Opsiyonel - Demo için güzel durur)
with st.expander("Veritabanı Tablolarını Gör"):
    conn = sqlite3.connect('magaza.db')
    st.write("Ürünler Tablosu:")
    st.dataframe(pd.read_sql("SELECT * FROM urunler", conn))
    st.write("Satışlar Tablosu:")
    st.dataframe(pd.read_sql("SELECT * FROM satislar", conn))
    conn.close()

```

Bunlar genel olarak başlangıç adımlarıydı. Sonrasında Google colab a geçiş yaptım ve uygulamayı orada tam anlamıyla çalışır hale getirdim ve indirilebilecek formu oluşturdum. Ayrıca bu aşamada OpenAI API yerine Google API 'a geçiş yaptım.

Sonrasında app dosyasını aldım. text_to_sql_app.py

Vs code uygulamasında copilot → Claude Opus (4.5) ile bu uygulamanın streamlit ile web tarafını oluşturdum.

"Uygulama'da web üzerinden CRUD işlemleri de gerçekleştirilebilsin ve Veri tabanı anlık ve güncel olarak harici bir sekmede düzenlenebilsin." promptu ile text_to_sql_web.py dosyasını oluşturdum. Sonrasında sistemi web üzerinden test ettim.

pip install pandas streamlit google-generativeai → Gerekli paketler

Ayarlar

Google AI Araştırma

Veritabanı Şeması

```
TABLE products
product_id (INTEGER)
product_name (TEXT)
category (TEXT)
price (REAL)

TABLE saless
sale_id (INTEGER), pk
product_id (INTEGER)
customer_id (INTEGER)
sale_date (TEXT)
quantity (INTEGER)
total_amount (REAL)
```

Örnek Sorgular

- Tüm ürünleri listele
- En pahalı 3 ürünü göster
- Toplam satış miktarını hesapla
- Kategori bazında ürün sayısını göster
- En çok satan 5 ürünü listele
- Tamamı ayırdıkları satılanları göster

Text-to-SQL & CRUD Uygulaması

AI Sorgu

Ürün Yönetimi

Satış Yönetimi

Yapay Zeka ile Doğal Dil Sorgusu

Sorgunuzu Yazın

Doğal dilde sorgunuzu yazın:

Örnek: En çok satan 5 ürünü listele

SQL Çıktısı

Formüle

Mevcut Veriler Önizleme

Ürünler

Satışlar

product_id	product_name	category	price
0	1 Laptop	Electronics	1200
1	2 Mouse	Electronics	25
2	3 Keyboard	Electronics	75
3	4 Monitor	Electronics	300
4	5 Desk Chair	Furniture	150

© Google Gemini AI ile desteklenmektedir | SQLite Veritabanı

AI Sorgu

Ürün Yönetimi

Satış Yönetimi

Satış Yönetimi

Satış Listesi

sale_id	product_name	customer_id	sale_date	quantity	total_amount
3	Laptop		103 2024-07-01	1	1200
4	Keyboard		101 2024-07-02	1	75
5	Monitor		104 2024-07-05	1	300
6	Desk Chair		105 2024-07-08	1	150
7	Coffee Mug		102 2024-07-10	3	30
8	Notebook		101 2024-07-12	5	25
9	Pen Set		108 2024-08-01	1	12
10	Laptop		104 2024-08-03	1	1200
11	Mouse		105 2024-08-05	1	25
12	Laptop		101 2025-12-03	3	3600

Yeni Satış Ekle

Satış Güncelle

Satış Sil

Yeni Satış Ekle

Ürün Seçin

Laptop (81.200.0)

Satış Tarihi

2025/12/04

Müşteri ID

101

Adet

1

Toplam Tutar: 81.200.00

AI Sorgu

Ürün Yönetimi

Satış Yönetimi

Ürün Yönetimi

Ürün Listesi

product_id	product_name	category	price
1	Laptop	Electronics	1200
2	Mouse	Electronics	25
3	Keyboard	Electronics	75
4	Monitor	Electronics	300
5	Desk Chair	Furniture	150
6	Coffee Mug	Kitchenware	10
7	Notebook	Stationery	5
8	Pen Set	Stationery	12

Yeni Ürün Ekle

Ürün Güncelle

Ürün Sil

Ürün Güncelle

Güncellenecek Ürünü Seçin

1 - Laptop

Ürün Adı

Laptop

Kategori

Electronics

Fiyat (₺)

1200,00

Güncelle

Yapay Zeka ile Doğal Dil Sorgusu

Sorgunuzu Yazın

Doğal dilde sorgunuzu yazın:

Tüm ürünleri listele

SQL Oluştur Temizle

Mevcut Veriler Önizleme

Ürünler Satışlar

product_id	product_name	category	price
0	1 Laptop	Electronics	1200
1	2 Mouse	Electronics	25
2	3 Keyboard	Electronics	75
3	4 Monitor	Electronics	300
4	5 Desk Chair	Furniture	150

Oluşturulan SQL Sorgusu

```
SELECT * FROM products;
```

Sorgu Sonuçları

8 satır bulundu

product_id	product_name	category	price
0	1 Laptop	Electronics	1200
1	2 Mouse	Electronics	25
2	3 Keyboard	Electronics	75
3	4 Monitor	Electronics	300
4	5 Desk Chair	Furniture	150
5	6 Coffee Mug	Kitchenware	10
6	7 Notebook	Stationery	5
7	8 Pen Set	Stationery	12

Yapay Zeka ile Doğal Dil Sorgusu

Sorgunuzu Yazın

Doğal dilde sorgunuzu yazın:

Toplam kaç tl kazandık satışlardan ?

SQL Oluştur Temizle

Mevcut Veriler Önizleme

Ürünler Satışlar

product_id	product_name	category	price
0	1 Laptop	Electronics	1200
1	2 Mouse	Electronics	25
2	3 Keyboard	Electronics	75
3	4 Monitor	Electronics	300
4	5 Desk Chair	Furniture	150

Oluşturulan SQL Sorgusu

```
SELECT SUM(total_amount) FROM sales;
```

Sorgu Sonuçları

1 satır bulundu

SUM(total_amount)
0