

## Prompt Tabanlı vs Knowledge Tabanlı SQL Query Parsing Karşılaştırması:

Bu çalışma, aday sorgularını SQL filtre koşullarına dönüştürmek için iki farklı yaklaşımı karşılaştırmaktadır:

- Prompt-tabanlı yaklaşım:** Tüm schema ve kurallar sistem prompt'una gömülür.
- Knowledge-tabanlı yaklaşım:** Schema ve kurallar bir bilgi tabanında (knowledge + vector DB) tutulur, model sadece retrieval ile bunlara erişir.

## Karşılaştırma Tablosu

Kriter	Prompt Tabanlı	Knowledge Tabanlı
<b>Doğruluk</b>	Basit koşulları yakalar (örn: age < 28, university). Mezuniyet gibi daha soyut kuralları atlayabilir.	Daha kapsamlı çıkarım yapar (örn: age < 28, university, education_status = mezun).
<b>Reasoning (akıl yürütme)</b>	Sınırlı: Prompt'taki kuralları harfiyen uygular.	Daha yoğun reasoning: Knowledge'den aldığı parçaları birleştirerek ek çıkarımlar yapabilir.
<b>Token Kullanımı</b>	Yüksek input token (prompt büyüdükçe artar).	Düşük input token (prompt sabit, knowledge retrieval yapar).
<b>Performans (Süre)</b>	Daha hızlı (tek adımda çalışır).	Daha yavaş (retrieval + reasoning + cevap).
<b>Bakım / Yönetim</b>	Yeni alan eklemek için prompt güncellemek gerekir (kod değişimi).	Yeni alan eklemek için sadece knowledge update yeterli (prompt sabit kalır).
<b>Maliyet</b>	Uzun vadede artar (her çağrıda tüm schema'yı gönderirsin).	Uzun vadede düşer (sabit prompt, sadece ilgili schema parçaları gelir).
<b>Ölçeklenebilirlik</b>	Küçük schema için uygun.	Orta/büyük schema ve dinamik alan yönetimi için uygun.
<b>Esneklik</b>	Katı kurallar → esneklik düşük.	Daha esnek, alanları genişlettikçe faydası artar.

## DeneySEL Gözlemler

**Sorgu:**

28 yaşından küçük ostim teknik üniversitesinden mezun adaylar

- **Prompt-tabanlı çıktı:**

```
{
  "conditions": [
    {"field": "age", "comparison": "less_than", "value": 28},
    {"field": "education/university", "comparison": "equal", "value": "ostim
teknik üniversitesi"}
  ]
}
```

→ “mezun” bilgisini atladi.

- **Knowledge-tabanlı çıktı:**

```
{
  "conditions": [
    {"field": "age", "comparison": "less_than", "value": 28},
    {"field": "university", "comparison": "equals", "value": "Ostim
Teknik Üniversitesi"},
    {"field": "education_status", "comparison": "equals", "value": "mezun"}
  ]
}
```

→ Yaş, üniversite ve mezuniyet durumunu eksiksiz çıkardı.

- **Prompt-tabanlı yaklaşım**, küçük schema’larda hızlı ve basit çözümler için uygundur. Ancak schema büyüdükçe prompt yönetimi zorlaşır, maliyet artar ve kuralların kapsayıcılığı sınırlı kalır.
- **Knowledge-tabanlı yaklaşım**, ek reasoning adımları sayesinde daha doğru ve kapsamlı sonuçlar verir. Yönetim ve ölçeklenebilirlik açısından çok daha avantajlıdır. Özellikle dinamik schema’lar ve kompleks sorgular için önerilir.

Özellik	Prompt-only	Knowledge tabanlı
Prompt uzunluğu	Çok uzun, karmaşık	Çok kısa, net
Schema güncelleme	Prompt edit → riskli	Sadece knowledge update
Doğruluk	Yanlış alanlar olabilir	Schema’ya sadık kalır
Performans / Token	Fazla token harcar	Az token (sadece retrieval)

Ölçeklenebilirlik

Zor

Kolay (yeni schema eklenir)