

# Çift Yıldız Kataloglarını Sorgulamak için Python Kütüphanesi

Sefa Saylan<sup>1\*</sup>,

<sup>1</sup>İstanbul Üniversitesi, Astronomi ve Uzay Bilimleri Bölümü, İstanbul

## Özet

Bu çalışmada "General Catalogue of Variable Stars", "Sixth Catalog of Orbits of Visual Binary Stars", "Ritter Cataclysmic Binaries Catalog (Rev. 7.21)" katalogları \*.sql formatına çevrilip mysql veri tabanına eklenmiştir. Geliştirilen bu veri tabanı ile çift yıldızlara ait her türlü bilgiyi sorgulayarak bir liste oluşturulacaktır. Oluşturulan python kütüphanesi yardımı ile ilk önce web sayfasına, web sayfası üzerinden de veri tabanına erişim sağlanacaktır. Bu sayede veri tabanının web servis olarak kullanılması mümkün hale gelecektir. Sonuç olarak astronomların çok kısa sürede istenilen veriye ulaşmaları hedeflenmektedir.

**Anahtar Kelimeler:** astronomical data bases: miscellaneous, Gözlemleri, Teleskoplar, Aletler, Yazılım

## 1 Giriş

Teknolojinin çağımızdaki etkisini gördüğümüz alanlardan birisi de astronomik veri miktarının her geçen saniye artmasıdır. Bu sonuçtan yola çıkarak çift yıldız kataloglarının sayısında da artış gözlemek mümkündür. Her bir katalogdaki veriye ulaşmak için katalogların web sayfasına girilmeli ve oradaki veri indirgenmelidir. Verilerin formatlı çıktısının verilmediği kataloglar göz önüne alındığında istenilen verilere ulaşmak için kaybedilen zaman miktarı oldukça artacaktır. Bu çalışmadaki amacımız istenilen veriye ulaşmak için harcanan zamanın minimuma indirgenmesidir.

## 2 Method ve Veri

Bu çalışmada php, python ve sql yazılım teknolojilerini kullanılmış ve veri olarak da çift yıldızlara ilişkin üç katalog kullanılmıştır. Önce [1][2][3] katalogları \*.sql formatına çevrildi daha sonra php dili yardımı ile her bir katalog bir tablo olarak veri tabanına aktarıldı.

Veri tabanının oluşturulabilmesi için kataloglar python dili içerisinde bulunan "split()" fonksiyonu yardımı ile parçalanarak \*.sql dosyası oluşturulup veri tabanına aktarılabilir duruma getirilmektedir. \*.sql dosyalarının büyüklüğünden dolayı bu aktarma işlemi phpmyadmin üzerinde bulunan Import bölümü ile gerçekleştirilemeyip php dili ile okutulup veri tabanına eklenmesi sağlandı.

[1] katalogunda veriler formatlı bir şekilde ayrılmış "—" ayrıca split fonksiyonunda kullanılarak gerekli çevrim kolay bir şekilde gerçekleştirilmiştir.

[2] katalogu formatlı olmayan bir katalog örneğidir. Bu katalogun çevrimi için kullanılan yöntem Kolon adlarının altında bulunan her bir satırda her bir kolon adının karakter sayısı kadar alınarak sütunlara ait veriler elde edilmiştir ve bu verilerle \*.sql çevrimi gerçekleştirilmiştir.

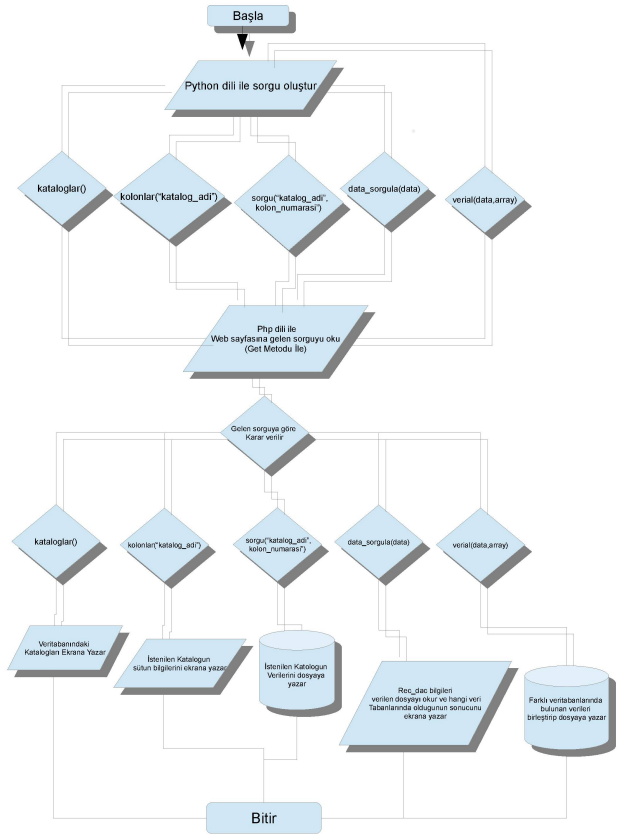
[3] Katalog da [1] kataloguyla aynı şekilde veri tabanına eklenmiştir.

Bu aşamaya kadar gerçekleşen süreçte veri tabanımız oluşturuldu. Şimdiki aşamada veri tabanını sorgulamak için php ve python arasındaki ilişki aşağıdaki akış şemasında gösterilmektedir.

Kullanıcının Python kütüphanesini kullanabilmesi için gerekli olan komutlar aşağıda tanımlanmıştır.

1	RA, Dec (J2000)...	WDS.....	DD.....	ADS.....	HD.....	HIP.....	V1.11*	V2.22*	PPPPP	PPPPPP	eee.eeeeee	AAA.AAAA*	ee.eeeeee
2	000000.91-192955.8	00000-1930	LTT 9831	.	224690	2	9.0	.	499.7989	d	18.8466	14.31	m 2.81
3													
4	000019.10-441726.0	00003-4417	I 1477	.	224750	25	6.80	7.56	384.1	y	.	1.023	a .
5	000019.10-441726.0	00003-4417	I 1477	.	224750	25	6.80	7.56	384.1	y	22.5	1.023	a 0.096
6													

Şekil 1. [2] Kataloguna ait verinin \*.sql formatına çevrilmiş örneği.



Şekil 2. Akış şeması

\* sefa.saylan@hotmail.com

```
import bdstpy as bs // Kütüphaneyi çalışılan dosyaya yükler.
bs.katalog() // Veri tabanında bulunan katalogların adını listeler.
bs.kolonlar('Katalogunadı'); // Çalışılmak istenilen katalogun hangi bilgileri içerdiğini gösterir.
bs.sorgu("GCSV","0" '1:10',2,3") // İstenilen katalogu bilgisayara kaydeder.
```

Burada açıklanması gereken nokta "0\_1:10',2,3" argümanıdır. Bu argüman yardımı ile her bir virgül kolon numarasını belirtmektedir ". \_1:10'" işlemi ise veriye istenilen limiti koymak için kullanılır. Bu komutun anlamı: "0= Id , 2 = GCVS , 3=RA\_Dec" olmak üzere.GCSV katalogunun içerisinde ID'si 1 ile 10 arasındaki dataların ID, GCVS , Ra\_Dec verilerini bilgisayara kaydeder.

```
bs.data" sorgula("dosya" yolu) // İçerisinde sağ açıklık ve dik açıklık bilgileri bulunan dosyayı
// veri tabanlarında sorgular ve hangi veri tabanlarında olduğu
// ile ilgili sonucu ekrana yazdırır.
bs.verial("dosya" yolu,
"Bulduğu birinci katalog:'istenilen sütunlar'
-Bulduğu ikinci veri tabanı:'istenilen sütunlar'")
```

Bu komut sayesinde farklı kataloglarda bulunan yıldızın istenilen verilerine eş zamanlı ulaşmak mümkün olacaktır

### 3 Hedefler

Çalışmanın daha da iyi sonuç verebilmesi için katalog sayısının artırılması ve sorgulama biçimlerinde yenilikler getirilmesi hedeflemektedir. Web sayfası üzerinden kullanıcıların da kendi profillerini oluşturmalarına ve ellerindeki yıldız kataloglarını kimse ile paylaşmadan sistemi kullanarak indirmeleri de hedefler arasındadır.

### Teşekkür

Çalışmamı gerçekleştirmem için destek ve danışmanlıkta bulunan Sayın Prof. Dr. M. Türker Özkan'a Python programlama dilini öğrenmem için tavsiyede bulunan Sayın Dr. Murat Gezer'e ve Fizikçiler ve Astronomlar için Bilgisayarla Hesaplama Teknikleri Yaz Okulu 'nu düzenleyen Bilim Akademisi Akademisyenlerine teşekkür ederim.

### Kaynaklar

Samus N.N.; Durlevich O.V.; et al General Catalogue of Variable Stars (Samus+ 2007-2011)  
The Sixth Catalog of Orbits of Visual Binary Stars (Henceforth Sixth Catalog ; Hartkopf et al. 2001a)  
Catalogue of cataclysmic binaries, low-mass X-ray binaries and related objects (Seventh edition) Ritter. H. Kolb.U, Release 7.21, March 2014.

### Erişim:

024-1810: [UAK-2015 Program](#) --- [UAK Bildiri](#) --- [Turkish J.A&A.](#)