

# Yapay Zeka Uzmanlık Programı

## Veri Yönetimi

2024

## Programlama Ödevi

---

### 1 Giriş

Bu ödev kapsamında, bir kitabevi için veritabanı oluşturmanız, tanımlanmış SQL sorgularını oluşturmanız ve çalıştırmanız istenmektedir. Bunun için belirli görevleriniz ve iyi tanımlanmış arayüzleriniz bulunmaktadır. Yapmanız gereken, verilen görevleri yerine getirmek için sağlanan arayüzleri uygulamaktır. Gerekli tüm veri dosyaları, sınıflar ve uygulamanız gereken arayüz kaynak dosyaları olarak sağlanmaktadır. Arayüzü grafik kullanıcı arayüzü (GUI) ile karıştırmayın. Bu ödev kapsamında herhangi bir GUI tasarlamayacaksınız. İlk yapmanız gereken, Bölüm 3'te verilen şemaya uygun gerekli tabloları oluşturan fonksiyonları tamamlamaktır. Daha sonra, ödevde verilen görevleri yerine getirmek için sorgular oluşturmalsınız. Son olarak, tanımlanan parametreler verildiğinde istenilen sonuçları veren sorguları kullanarak arayüzü uygulamalsınız. Tüm değişikliklerinizi bookdb.py dosyasında yapacaksınız, diğer dosyalarda değişiklik yapmayacaksınız. Göreviniz sağlanan size sağlanan IBOOKDB interface'ini BOOKDB sınıfı içinde gerçekleştirmektir. Kendi değerlendirmenizi evaluation.py dosyasını çalıştırarak yapabilirsiniz. Asıl değerlendirme bizim tarafımızdan, kendi bilgisayarımızda yapılacaktır.

### 2 Objectives

Bu ödevin amacı, sizi aşağıdaki konularda bilgilendirmeyi amaçlamaktadır:

- Python programlama dilinin temelleri,
- MySQL Sunucusuna bağlanma, sorgulama ve Python kullanarak DML'leri uygulama

### 3 Şema

Bu ödev kapsamında aşağıda verilen şema kullanılacaktır.

**author**(author\_id:int, author\_name:varchar(60))

**publisher**(publisher\_id:int, publisher\_name:varchar(50))

**book**(isbn:char(13), book\_name:varchar(120), publisher\_id:int, first\_publish\_year:char(4), page\_count:int, category:varchar(25), rating:float) **REFERENCES** publisher(publisher\_id)

**author\_of**(isbn:char(13),author\_id:int) **REFERENCES** book(isbn) author(author\_id)

**phw1**(isbn:char(13), book\_name:varchar(120), rating:float)

Göreviniz, IBOOKDB arayüzünü uygulayan BOOKDB adında bir sınıf oluşturmaktır (bookdb.py dosyasında) BOOKDB sınıfının aşağıdaki görevleri yerine getirebilmesi gerekmektedir:

- Veritabanı tablolarını oluşturma
- Tablolara veri ekleme
- Verilen sorular için gereken SQL sorgularını alabilme

- Gerekli DML işlemlerini gerçekleştirme
- Veritabanı tablolarını silme

Görevler aşağıda daha detaylı olarak açıklanmıştır.

Her görev için IBOOKDB arayüzünde karşılık gelen bir yöntem bulunmaktadır. Bu yöntemleri BOOKDB sınıfında uygulamanız gerekmektedir. Bu görevleri yerine getirmek için gerekli veri dosyaları (tabloları doldurmak için) sağlanacaktır. **data** klasöründe, her bir tabloya karşılık gelen 4 txt dosyası bulunmaktadır. Veri eklerken bu tabloları kullanacaksınız. Bu görevleri yerine getirmek için veri dosyaları, arayüzler ve sınıflar sağlanacaktır. Tüm bilgilerin eksiksiz ve tutarlı olacağını varsayabilirsiniz, yani bir sorguyu çalıştırmadan önce gerekli tüm veriler eklenmiş olacaktır. Sağlanan kaynak dosyalarda fonksiyonların kullanımı hakkında detaylı açıklamaları bulabilirsiniz. Tabloları oluştururken **foreign keys** (yabancı anahtarlar) tanımlamayı unutmayın. Lütfen MySQL sorgularınızda lazım olduğunda DISTINCT anahtar kelimesini kullanmayı unutmayın.

### 3.1 Veritabanı Tablolarını Oluşturma (create) (10 puan)

```
createTables(self):
```

Yukarıda tanımlanan şemaya göre tüm tabloları oluşturacaksınız (create).

Diğer herhangi bir veritabanı işlemi gerçekleştirilmeden önce tabloların oluşturulacağını varsayabilirsiniz.

Bu fonksiyon başarıyla oluşturulan tablo sayısını döndürmelidir.

### 3.2 Tablolara Veri Ekleme (5 puan)

```
insertAuthor(self, authors: list[Author]) -> int:
insertBook(self, books: list[Book]) -> int:
insertPublisher(self, publishers: list[Publisher]) -> int:
insertAuthor_of(self, author_ofs: list[Author_of]) -> int:
```

Uygun tablolara veri ekleyeceksiniz.

Bu fonksiyon başarıyla eklenen satır sayısını döndürür. Bu görevde, lütfen **phw1** tablosuna herhangi bir veri eklemeyin.

### 3.3 Sorgu 1 (5 puan)

```
functionQ1(self) -> list[QueryResult.ResultQ1]:
```

En fazla sayfaya sahip olan kitabın(kitapların) isbn, first\_publish\_year, page\_count ve publisher\_name bilgilerini listeleyeceksiniz.

Sonuçları isbn'ye göre artan sırada sıralamalısınız.

### 3.4 Sorgu 2 (5 puan)

```
functionQ2(self, author_id1: int, author_id2: int) -> list[QueryResult.ResultQ2]:
```

Verilen iki yazarın **birlikte** yazdığı kitapları yayımlayan yayınevlerinin; publisher\_id'lerini ve bu yayınevlerinin yayımladığı tüm kitapların ortalama sayfa sayısını (page\_count) listeleyeceksiniz.

Sonuçları publisher\_id'ye göre artan sırada sıralamalısınız.

### 3.5 Sorgu 3 (10 puan)

```
functionQ3(self, author_name: str) -> list[QueryResult.ResultQ3]:
```

Verilen author\_name'e sahip yazar(lar)ın en erken yayımlanan kitabının(kitaplarının) book\_name, category ve first\_publish\_year bilgilerini listeleyeceksiniz.

Sonuçları book\_name, category ve first\_publish\_year'a göre artan sırada sıralamalısınız.

### 3.6 Sorgu 4 (10 puan)

```
functionQ4(self) -> list[QueryResult.ResultQ4]:
```

En az 3 kelime içeren (örnek: "Koc Üniversitesi Yayınları") isimlere sahip, en az 3 kitap yayımlamış olan ve tüm kitaplarının ortalama rating'i 3'ten büyük (>) olan yayınevlerinin publisher\_id'lerini ve yayınladıkları farklı kategorileri (category) listeleyeceksiniz. Not: publisher\_name içindeki her kelimenin bir boşlukla ayrıldığı varsayılabilir.

Sonuçları publisher\_id ve category'e göre artan sırada sıralamalıyız.

### 3.7 Sorgu 5 (10 puan)

```
functionQ5(self, author_id: int) -> list[QueryResult.ResultQ5]:
```

Verilen author\_id'nin çalıştığı **tüm yayınevleriyle** çalışmış yazarların author\_id ve author\_name bilgilerini listeleyeceksiniz.

Sonuçları author\_id'ye göre artan sırada sıralamalıyız.

### 3.8 Sorgu 6 (10 puan)

```
functionQ6(self) -> list[QueryResult.ResultQ6]:
```

'Seçici' yazarların author\_id, isbn(ler)ini listeleyin. "Seçici" yazarlar; yalnızca kendi kitaplarını yayımlayan yayınevleriyle çalışmış olan yazarlardır. (yani, farklı yazarların kitaplarını yayınlamamışlardır).

Sonuçları author id ve ISBN numarasına göre artan sırada sıralamalıyız.

### 3.9 Sorgu 7 (10 puan)

```
functionQ7(self, rating: float) -> list[QueryResult.ResultQ7]:
```

'Roman' kategorisinde en az 2 kitap yayımlamış olan ve kitaplarının ortalama rating'i verilen değerden büyük (>) olan yayınevlerinin publisher\_id ve publisher\_name'lerini listeleyeceksiniz.

Sonuçları publisher\_id'ye göre artan sırada sıralamalıyız.

### 3.10 Sorgu 8 (Toplu(Bulk) Ekleme) (10 puan)

```
functionQ8(self) -> list[QueryResult.ResultQ8]:
```

- Mağazadaki bazı kitaplar birden fazla kez yayımlanmış olabilir: aynı isimlere (book\_name) sahip olmalarına rağmen, farklı isbn'lerle yayımlanmış olabilirler. Bu kitapların her biri için, en küçük rating'e sahip olanlarının isbn, book\_name ve rating'lerini bulun ve bunları tek bir BULK insert sorgusu kullanarak **phw1** tablosuna kaydedin. Eğer en düşük puanlı birden fazla kitap varsa, o zaman hepsini kaydedin.
- Toplu ekleme işleminden sonra, **phw1** tablosundaki tüm satırların isbn, book\_name ve rating'lerini listeleyiniz. Isbn'e göre artan sırada sıralamalıyız.

### 3.11 Sorgu 9 (Update) (5 puan)

```
functionQ9(self, keyword: str) -> float:
```

- İsimlerinin (book\_name) içerisinde verilen anahtar kelime(**keyword**) olan kitapların rating'lerini bir (1) artırın. Maksimum rating 5'ten fazla olamaz, bu nedenle değiştirmeden önceki rating 4'ten büyükse, o kitabın rating'ini güncellemeyiniz.
- Güncelleme işleminden sonra, tüm kitapların rating'lerinin toplamını döndürünüz.

### 3.12 Sorgu 10 (Delete) (5 puan)

```
function10(self) -> int:
```

- Henüz hiç kitap yayınlamamış yayınevlerini publisher tablosundan siliniz.
- Silme işleminden sonra, yayınevleri tablosundaki kayıtların sayısını döndürünüz.

### 3.13 Veritabanı tablolarını silme (drop table) işlemi. (5 puan)

Oluşturduğunuz (create) tüm tabloları siliniz (drop)  
Başarıyla silinen (drop edilen) tablo sayısını döndürünüz.

## 4 Kurallar

1. Programlama dili: Python.
2. Kodunuzu göndermeden önce kendi bilgisayarınızda ve MYSQL sunucusunda test etmeniz gerekmektedir. Ancak değerlendirme aşaması kendi bilgisayarım ve MYSQL sunucum üzerinde yapılacaktır.
3. Girdi: Metin olan tüm girişler büyük/küçük harfe duyarlı olacak ve herhangi bir yabancı karakter içermeyecektir.
4. Ödevin geç yüklenmesi kabul edilme-mektedir.
5. Kopya çekme konusunda sıfır tolerans politikamız bulunmaktadır. Kopya çekenler sıfır not alacaktır.
6. Değerlendirme: Size sağlanan veri kümesinden daha büyük ve benzer veriler kullanacaktır. Ödeviniz otomatik olarak "black-box" teknik kullanılarak değerlendirileceğinden, ödev şartlarına kesinlikle uyulmalıdır.

## 5 Ödev Yükleme

Kullanıcı yalnızca **bookdb.py** dosyasını gönderecektir. bookdb.py dosyasını aşağıdaki formatta zip formatında sıkıştırdıktan sonra göndermelisiniz.

ad\_soyad.zip

Yüklemeyi şu linkten. yapacaksınız

## 6 Yararlı linkler

- MySQL kurulumu:  
<https://dev.mysql.com/downloads/installer/>
- MySQL Reference Manual:  
<http://dev.mysql.com/doc/refman/8.0/en/>
- Temel MySQL Eğitimi:  
<http://www.mysqltutorial.org/basic-mysql-tutorial.aspx>

*Ramazan Bayramınız en iyi dileklerle kutluyorum,*