

**Veriliş Tarihi: 28/02/2020**

**Teslim Tarihi: 13/02/2020 23:59**

**Teslim Şekli: bil211spring2020@gmail.com adresine hw3\_isim\_soyisim.zip şeklinde yollayabilirsiniz**

### **Soru 1)(100 Puan)**

Düzenli verileri CSV dosyalarında tutmak yaygın bir kullanım şeklidir. Her satırda ayrı bir veri bulunmakta ve her sütunda ise ayrı bir veri türü bulunmaktadır. Örnek bir dosya aşağıda gösterilmiştir.

PLAYER	POSITION	TEAM
Paul Millsap	PF	Atlanta Hawks
Al Horford	C	Atlanta Hawks
Tiago Splitter	C	Atlanta Hawks
Jeff Teague	PG	Atlanta Hawks
Kyle Korver	SG	Atlanta Hawks

Bu dosyada NBA oyuncularına dair bilgiler verilmektedir. Her bir satır başka bir oyuncuya ait olup, ilk sütun ismini, ikinci sütun oynadığı pozisyonu, ve 3. sütun takımı göstermektedir.

Bu tip CSV dosyalarındaki verileri okuyup, veriler üzerinde çeşitli işlemler yapabileceğimiz generic bir sınıf yazmanız gerekmektedir. CSV dosyalarında ilk satırda her zaman veri türü isimleri olacak (Player, position gibi.) CSV dosyalarında her zaman aynı veri türü olacağını varsayabilirsiniz. Eğer yoksa, aşağıda constructor metodunda belirtildiği gibi hata atılmalıdır ve işlem yapılmamalıdır.

Database adında bir sınıf yazacaksınız. Metotları şöyledir:

**Database (String dosya\_ismi):** Constructor metodu. Verilen csv dosyasındaki verileri yükler. Kullanıcının belirttiği veri türünde tüm sütunları tutar. Integer ise Integer, String ise String. Eğer dosyada bu duruma uymayan bir veri varsa, hata atar ve nesneyi oluşturmaz.

**Database where(String column\_name, T value):** column\_name olarak belirtilen sütunda value ile aynı değerde olan tüm satırları içeren **yeni** bir Database nesnesi dönmeli. Asıl veritabanının içeriği değiştirilmemeli. Örneğin, yukarıdaki tabloyu tutan bir Database nesnesinde where("POSITION","C") çağrısı şu nesneyi dönmeli:

PLAYER	POSITION	TEAM
Al Horford	C	Atlanta Hawks
Tiago Splitter	C	Atlanta Hawks

**Database select(String[] column\_names):** column\_names dizisinde belirtilen sütunları içeren **yeni** bir Database nesnesi dönmeli. Asıl veritabanının içeriği değiştirilmemeli. Örneğin, yukarıdaki örnek tabloyu içeren Database nesnesine select({"POSITION", "TEAM"}) çağrısı yapıldığında şu tabloyu dönmeli.

POSITION	TEAM
PF	Atlanta Hawks
C	Atlanta Hawks
C	Atlanta Hawks
PG	Atlanta Hawks
SG	Atlanta Hawks

**void add\_column(String label, T[] data):** Veritabanına verilen isimde yeni bir sütun ekler. Eğer parametre olarak verilen data dizisinin boyutu veritabanındaki veri sayısından fazla veya az ise hata mesajı yazar ve ekleme yapmaz. Örneğin, ilk örnekteki veritabanı için add("BirthState", {"OH", "NY", "CA", "CA", "IL"}) çağrısı sonucu veri tabanı şu halde olmalı:

PLAYER	POSITION	TEAM	BirthState
Paul Millsap	PF	Atlanta Hawks	OH
Al Horford	C	Atlanta Hawks	NY
Tiago Splitter	C	Atlanta Hawks	CA
Jeff Teague	PG	Atlanta Hawks	CA
Kyle Korver	SG	Atlanta Hawks	IL

**void remove(String column\_name, T value):** column\_name olarak belirtilen sütunda value ile aynı değerde olan tüm satırları silmeli. Örneğin, yukarıdaki tabloyu tutan bir Database nesnesinde remove("POSITION","C") çağrısı sonucu tablo şu halde olmalı.

PLAYER	POSITION	TEAM
Paul Millsap	PF	Atlanta Hawks
Jeff Teague	PG	Atlanta Hawks
Kyle Korver	SG	Atlanta Hawks

**void display():** Veritabanındaki tüm verileri ekrana yazdırır. En başta sütun isimleri olmalı. Sütunlar arasına tab karakteri koyulmalı.

**Database join(Database second, String column\_name):** İki veri tabanını belirtilen sütuna göre birleştirir ve sonucu yeni bir Database nesnesi olarak döner. Birleştirilen veritabanlarının içeriği değiştirilmez. Birleştirme işleminde belirtilen sütunda aynı değeri olan satırlar birleştirilir. Örneğin, X nesnesi şöyle olsun:

ID	Name
1	A
2	B
3	C

Y nesnesi de şöyle olsun.

ID	Surname
1	D
2	E
4	E

O zaman `x.join(Y, "ID")` şu sonucu dönmeli

ID	Name	Surname
1	A	D
2	B	E

Yani, ikisinde de aynı değer olduğunda birleştirme olmalı. Eğer belirtilen `column_name` iki tablonun herhangi birinde yoksa, hata mesajı dönmeli ve ekleme yapılmamalı. Eğer birleştirme yaparken, eğer aynı isimde iki sütun olma durumu olursa, (parametre olarak verilen 2. Tablonun sütun ismine "2" eklenmeli. Örneğin, X tablosunda ID ve NAME sütunları olsun ve Y tablosunda da ID ve NAME sütunları olsun. `x.join(y,"ID")` olduğunda dönen tabloda ID, NAME ve NAME2 adında 3 sütun olmalı. Birleştirme uygulanan sütunda aynı değerden birden fazla olmayacağını varsayabilirsiniz. Örneğin, ID sütunlarında birden fazla 1, 2, 3 sayıları geçmeyecektir.