

## Veritabanı Yönetim Sistemleri Dersi Vize Sınavı

Ad Soyad:

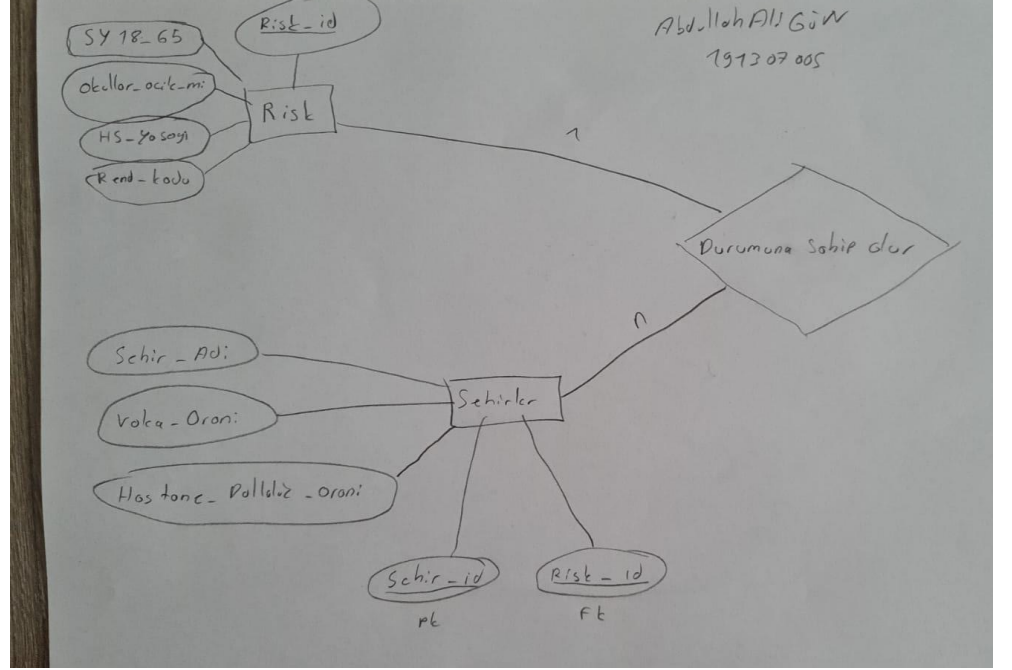
12.04.2021 - 19.04.2021

No:

Vize sınavı için her öğrencinin nCovid-19 (Yeni Korona Virüs Hastalığı) ile ilgili bir veri tabanı yazılımı üretmesi beklenmektedir. Üretmiş olduğunuz soruda veritabanı ilişkisel olmalıdır. Sizlerden beklentim aşağıda yer alan kurallara uygun olarak belgeyi hazırlamanızdır. Her maddenin altına ilgili cevapları yazınız. Ekran görüntüsü, fotoğraf vb koyabilirsiniz. Görüntüler düzgün olmalıdır. Dosya adı, ad\_soyad\_ogrencinumarası\_VTYS\_vize.docx (odt uzantılıda olabilir) ve ad\_soyad\_ogrencinumarası\_VTYS\_vize.pdf şeklinde olmalıdır. (Hem pdf dosyası hemde dosyanın Orijinal hali sisteme yüklenmelidir.) Ayrıca, tüm soruları cevapladıktan sonra, sisteme veritabanı yedeğininide yükleyiniz.

1. Yazılım ile ilgili problemi okuyan kişi net bir şekilde anlamalıdır. (Kısacası soruyu siz üretin.)
  - Türkiye Covid-19 risk haritası.
2. Problemde istenilenleri ortaya net bir şekilde koyunuz. (Yazdıklarınızı okuyan bir kişi sizin ney yapmak istediğinizi anlamalıdır.)
  - Şehir tablosunda şehir numarası, şehir adı, vaka oranı, hastane doluluk oranı ve risk numarası tutulacak.
  - Risk tablosunda risk numarası, hafta sonu sokağa çıkma yasağı var mı? yok mu? , sokağa çıkma yasağı saatleri, okullar açık mı? vb bilgileri tutacak.
  - Hastane tablosunda hastane adı, hastane numarası, şehir numarası, hastane doluluk oranı bilgileri tutulacak.
  - Şehirler Vaka sayılarına ve hastane doluluk oranlarına göre risk kodu alacaklar.  
Riskli Bölge(1) %10 altında hastane doluluk oranı veya %5 altında vaka oranı  
Orta Riskli Bölge(2) %6-10 arası vaka oranı  
Riskli Bölge %11-15 arası vaka oranı  
Çok Riskli Bölge %70 in üstüne en az 1 hastane doluluk oranı veya %16 üstü vaka oranı
3. Veritabanına ait kavramsal modelini çıkarınız. (Hangi varlıklarınız var? Bu varlıklara ait nitelikler nelerdir?)
  - Varlık : Şehir
    - Öznitelik : Şehir\_ID
    - Nitelik : Şehir adı
    - Nitelik : Vaka oranı
    - Nitelik : Hastane Doluluk
    - Yabancı Nitelik : Risk numarası
  - Varlık : Risk
    - Öznitelik : Risk\_ID
    - Nitelik : Renk kodu
    - Nitelik : HS(Hafta Sonu Sokağa Çıkma) yasağı
    - Nitelik : okullar\_acik\_mi
    - Nitelik : SY18\_65 (18 altı 65 üstü HS var mı)

4. Varlık İlişki Modelini (VİM) kağıda çizerek hazırlayınız. (Sorunun altına çizdiğiniz VİM fotosunu yapıştırıp açıklayınız.)



5. Kağıda çizdiğiniz VİM modelini draw.io ile hazırlayınız. (Sorunun altına hazırladığınız draw.io görüntüsünü ekleyiniz.)



6. Oluşturduğunuz VİM'e göre tablolara geçişi sağlayınız. (Unutanlar ders notu örneklerini incelesin. Tekrar söylüyorum, VİM'e göre tablolara geçiniz. Sorunun altına yazınız.)

- **Sehir**
- **Risk**
- Sehir(Sehir\_ID,Sehir\_Adi, Vaka\_Oranı, Hastane\_Doluluk\_Oranı)
- Risk(Risk\_ID,Renk\_Kodu,HS\_Yasagi,Okullar\_Acik\_Mı, SY18\_65)
- **Sehir** sahip olur **Risk** (durumuna)
- Sehir(Sehir\_ID,Sehir\_Adi, Vaka\_Oranı, Hastane\_Doluluk\_Oranı, Risk\_ID)
- Risk(Risk\_ID,Renk\_Kodu,HS\_Yasagi,Okullar\_Acik\_Mı, SY18\_65)

7. Her tabloda bulunan alanların veri tiplerini ve kısıtlamalarını belirleyiniz.

- risk (
  - risk\_ID int primary key not null,
  - renk\_kodu varchar(10) not null,
  - HS\_yasagi bool not null,
  - okullar\_acik\_mi bool not null,
  - SY18\_65 bool not null,
  - primary key(risk\_ID)
- sehirler(
  - sehir\_ID int primary key not null,
  - sehir\_Adi varchar(20) not null,
  - vaka\_Orani int not null,
  - hastane\_Doluluk\_Orani int not null,
  - risk\_ID int not null,
  - primary key(sehir\_ID),
  - FOREIGN KEY (risk\_ID) REFERENCES risk(risk\_ID)
  - constraint vokisit check(vaka\_Orani > -1 and vaka\_Orani <101),
  - constraint hdokisit check(hastane\_Doluluk\_Orani > -1 and hastane\_Doluluk\_Orani <101)

8. Veritabanı ve tabloları yaratan sql cümleciklerini (MySQL Workbench kullanarak) yazınız.

```
create database nCovid_19_191307005;
```

```
use nCovid_19_191307005;
```

```
create table risk (
```

```
    risk_ID int primary key not null,
```

```
    renk_kodu varchar(10) not null,
```

```
    HS_yasagi bool not null,
```

```
    okullar_acik_mi bool not null,
```

```
    SY18_65 bool not null,
```

```
    primary key(risk_ID)
```

```
);
```

```
create table sehirler(
```

```
    sehir_ID int primary key not null,
```

```
    sehir_Adi varchar(20) not null,
```

```
    vaka_Orani int not null,
```

```
    hastane_Doluluk_Orani int not null,
```

```
    risk_ID int not null,
```

```
    primary key(sehir_ID),
```

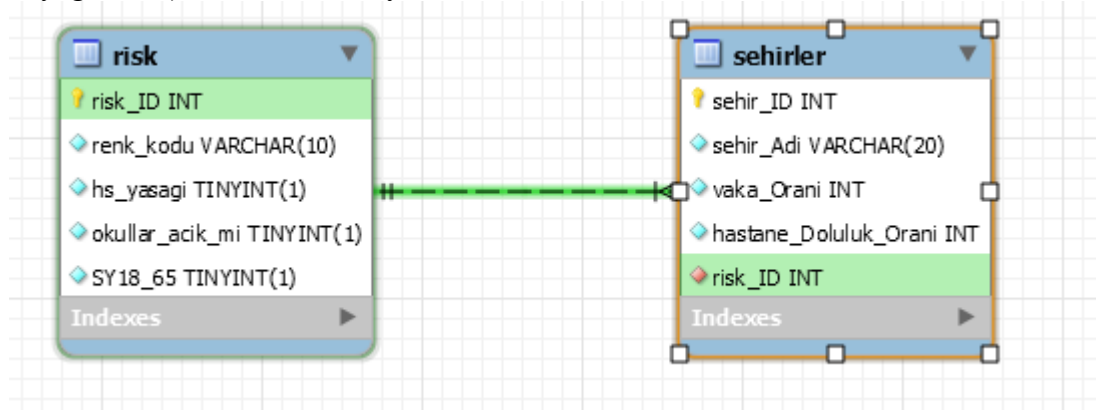
```
    FOREIGN KEY (risk_ID) REFERENCES risk(risk_ID)
```

```
    constraint vokisit check(vaka_Orani > -1 and vaka_Orani <101),
```

```
    constraint hdokisit check(hastane_Doluluk_Orani > -1 and hastane_Doluluk_Orani <101)
```

```
);
```

9. Yazdığımız sql cümlelerine göre oluşan ilişisel veritabanının, MySQL Workbenchte yer alan VİM (ER Diyagramını) bu bölüme ekleyiniz.



10. Veritabanında yeralan tüm tablolara bilgi girişi yapınız, sql cümleciklerini sorunun altına ekleyiniz.

```
INSERT INTO risk (risk_ID, renk_kodu, hs_yasagi, okullar_acik_mi, SY18_65)
```

```
VALUES (1, "yeşil", false, true, false),
```

```
(2, "sarı", false, true, true),
```

```
(3, "turuncu", true, true, false),
```

```
(4, "yeşil", true, false, false);
```

```
INSERT INTO sehirler (sehir_ID, sehir_Adi, vaka_Orani,
```

```
    hastane_Doluluk_Orani, risk_ID)
```

```
VALUES (1, "adana", 6, 4, null),
```

```
(6, "ankara", 8, 10, null),
```

```
(16, "bursa", 14, 4, null),
```

```
(41, "kocaeli", 18, 50, null),
```

```
(34, "istanbul", 11, 90, null),
```

(35, "izmir", 15, 11,null),  
(65, "van", 1,90,null);

11. Oluşturduğunuz veri tabanı üzerinde sorgulama, güncelleme ve silme işlemlerini yapınız. Cümleciği sorunun altına ekleyiniz. Tablodaki son durumun ekran görüntüsünü alarak uyguladığınız sql cümleciğinin altına ekleyiniz. (En az 9 adet sorgu olsun. Güncelleme ve silmeden sonra tablonun son durumunu gösteren seçme işlemi uygulanmalıdır. Güncelleme + seçme işlemi 1 sorgu kabul edilir. Çokta basit sorgular yazmayınız. Yazacağınız sorguya ait ilk olarak soru, sonra sql cümlesi son olarak ekran görüntüsü yer almalıdır.)

- Tablolardaki tüm şehirlerin vaka ve hastane oranlarına bakarak risk idlerinin güncellenmesi ( hocam tablolar vs kolay ve sade geldiği için gözüme sorgularda baya bir araştırma yapıp emek verildiğini hissettirmek için bunda baya uğraştım)

/\* kullanım için örnek kaynak :<https://github.com/ibrahimyldz11q/MYSQL-IF-ELSE/blob/master/ifelse.sql>\*/

DELIMITER //

CREATE PROCEDURE test(IN vaka int,IN hastane int, OUT durum int)

BEGIN

IF hastane < 10 THEN SET durum = 1;

ELSEIF hastane > 70 THEN SET durum = 4;

ELSEIF vaka < 6 THEN SET durum = 1;

ELSEIF vaka > 5 and vaka < 11 THEN SET durum = 2;

ELSEIF vaka > 10 and vaka < 16 THEN SET durum = 3;

ELSEIF vaka > 15 THEN SET durum = 4;

ELSE SET durum = null;

END IF;

END//

DELIMITER ;

/\*kullanım için örnek kaynak : <https://www.yusufsezer.com.tr/mysql-while/>\*/

DELIMITER //

CREATE PROCEDURE riskgüncelle()

BEGIN

DECLARE x INT;

SET x = 1;

WHILE x <= 7 DO

SET @riskid = 0;

CALL test((Select vaka\_Orani from sehirler where sehir\_ID = x),(Select hastane\_Doluluk\_Orani from sehirler where sehir\_ID = x),@riskid);

SELECT (@riskid);

UPDATE sehirler

SET risk\_ID = @riskid

WHERE sehir\_ID = x;

SET x = x + 1;

END WHILE;

END //

DELIMITER ;

CALL riskgüncelle();

Öncesinde tüm risk\_ID ler null değerdayken sonrası

|   | sehir_ID | sehir_Adi | vaka_Orani | hastane_Doluluk_Orani | risk_ID |
|---|----------|-----------|------------|-----------------------|---------|
| ▶ | 1        | adana     | 6          | 4                     | 1       |
|   | 2        | ankara    | 8          | 10                    | 2       |
|   | 3        | bursa     | 14         | 4                     | 1       |
|   | 4        | kocaeli   | 18         | 50                    | 4       |
|   | 5        | istanbul  | 11         | 90                    | 4       |
|   | 6        | izmir     | 15         | 11                    | 3       |
|   | 7        | van       | 1          | 90                    | 4       |
| * | NULL     | NULL      | NULL       | NULL                  | NULL    |

- Okulların kapalı olduğu şehirler  
select sehir\_Adi from sehirler where risk\_ID = (select risk\_ID from risk where okullar\_acik\_mi = false);

| Result Grid | Filter |
|-------------|--------|
| sehir_Adi   |        |
| ▶ kocaeli   |        |
| istanbul    |        |
| van         |        |

- Vaka oranının en düşük olduğu şehir/ler  
select sehir\_Adi from sehirler where min(vaka\_Orani);

| Result Grid | Filter Rows: |
|-------------|--------------|
| sehir_Adi   |              |
| ▶ van       |              |

- Hastane doluluk oranının en fazla olduğu şehir/ler  
select sehir\_Adi from sehirler where max(hastane\_Doluluk\_Orani);

| Result Grid | Filter R |
|-------------|----------|
| sehir_Adi   |          |
| ▶ istanbul  |          |
| van         |          |

- 7 numaralı şehrin hastane doluluk oranını 20 yapıp risk haritasını güncelle(1. Sorgu)  
UPDATE sehirler SET hastane\_Doluluk\_Orani = 20 WHERE sehir\_ID = 7;  
CALL riskguncelle();  
SELECT \* FROM sehirler;

Result Grid

Filter Rows:

Edit:

Export/Import:

|   | sehir_ID | sehir_Adi | vaka_Orani | hastane_Doluluk_Orani | risk_ID |
|---|----------|-----------|------------|-----------------------|---------|
| ▶ | 1        | adana     | 6          | 4                     | 1       |
|   | 2        | ankara    | 8          | 10                    | 2       |
|   | 3        | bursa     | 14         | 4                     | 1       |
|   | 4        | kocaeli   | 18         | 50                    | 4       |
|   | 5        | istanbul  | 11         | 90                    | 4       |
|   | 6        | izmir     | 15         | 11                    | 3       |
|   | 7        | van       | 1          | 20                    | 1       |
| * | NULL     | NULL      | NULL       | NULL                  | NULL    |

- 7 numaralı şehri silme şehir silme  
DELETE FROM sehirler WHERE sehir\_ID = 7;

|   | sehir_ID | sehir_Adi | vaka_Orani | hastane_Doluluk_Orani | risk_ID |
|---|----------|-----------|------------|-----------------------|---------|
| ▶ | 1        | adana     | 6          | 4                     | 1       |
|   | 2        | ankara    | 8          | 10                    | 2       |
|   | 3        | bursa     | 14         | 4                     | 1       |
|   | 4        | kocaeli   | 18         | 50                    | 4       |
|   | 5        | istanbul  | 11         | 90                    | 4       |
|   | 6        | izmir     | 15         | 11                    | 3       |
| * | NULL     | NULL      | NULL       | NULL                  | NULL    |

- 3 numaralı risk şehirlerinde hafta sonu sokağa çıkma yasağını kaldır  
update risk set hs\_yasagi = false where risk\_ID = 3;  
SELECT \* FROM risk;

|   | risk_ID | renk_kodu | hs_yasagi | okullar_acik_mi | SY18_65 |
|---|---------|-----------|-----------|-----------------|---------|
| ▶ | 1       | yeşil     | 0         | 1               | 0       |
|   | 2       | sarı      | 0         | 1               | 0       |
|   | 3       | turuncu   | 0         | 1               | 0       |
|   | 4       | yeşil     | 1         | 0               | 1       |
| * | NULL    | NULL      | NULL      | NULL            | NULL    |

- Hafta sonu sokağa çıkma yasağı olan şehirler  
select sehir\_Adi from sehirler where risk\_ID = (select risk\_ID from risk where  
hs\_yasagi = true);

|   | sehir_Adi |
|---|-----------|
| ▶ | kocaeli   |
|   | istanbul  |

- Kaç şehirde okullar kapalı değil  
select count(sehir\_Adi) from sehirler where risk\_ID != (select risk\_ID from risk  
where okullar\_acik\_mi = false);

|   | count(sehir_Adi) |
|---|------------------|
| ▶ | 4                |

## NOT:

- Öğrenciler bireysel olarak çalışmalıdır. Benzer veya aynı özelliklere sahip ödevler sisteme yüklenirse, ilgili tüm öğrenciler vize sınavından 0 (sıfır) alır.
- Sisteme yüklenecek dosyalar;
  - ad\_soyad\_ogrencinumarasi\_VTYS\_vize.docx (odt uzantılıda olabilir)
  - ad\_soyad\_ogrencinumarasi\_VTYS\_vize.pdf
  - nCovid-19\_ogrencinumarasi.sql
  - nCovid-19\_ogrencinumarasi.drawio