

A- Aşağıda verilen ilişkisel veritabanını kullanarak SQL sorgularında boş yerleri doldurunuz.

mesaj (mesaj_id, konu, icerik)

kime(mesaj_id, eposta, tarih)

musteri (eposta, isim, aile_nufusu, adres)

1- “Mutlu Yıllar” konulu mesaj gönderen tüm müşterilerin isim ve e-postalarını bulunuz.

```
select _musteri.eposta, isim___  
from mesaj, kime, muster  
where _konu='Mutlu Yıllar' AND  
        mesaj.mesaj_id=kime.mesaj_id AND  
        kime.eposta=musteri.eposta
```

2- Aile nüfusu 2 olduğunu söyleyen ve aynı adreste yaşayan tüm müşterilerin isimlerini ve adreslerini bulunuz.

```
select C1.isim, _C2.isim___, C1.adres  
from _musteri C1, muster C2  
where _c1.isim<>c2.isim___ AND, _c1.aile_nufusu=2_____ AND  
C2.aile_nufusu=2 AND _c1.adres=c2.adres___
```

- 3- Aile nüfusu ortalama aile nüfusunun en az %50 'sinden fazla olan tüm müşterileri ve aile nüfuslarını listeleyiniz.

```
create view ortalama_nufus (ort_nufus) as  
select _avg(aile_nufusu)_____  
from _musteri_____
```

```
select isim, aile_nufusu  
from _musteri_____ortalama_nufus  
where _aile_nufusu >= 1.5* ort_nufus _____  
order by _aile_nufusu_____
```

Aşağıda verilen ilişkisel şemayı kullanarak SQL sorgularında boş yerleri doldurunuz.

musteri(no, isim, yas, cinsiyet)
ucus(no, ucusNo, ucusTarihi)
ucusBilgi(ucusNo, nereden, nereye, ucusZamanı, sure)

Yukardaki veritabanı bir seyahat acentesine kayıtlı müşterileri göstermektedir. Müşteriler uçuş veya başka bir maksat (tatil vb) amacı ile veritabanına kayıt yapmış olabilirler. Her uçuşun hergün için yapıldığını varsayınız.

1. 10 yaşından büyük müşterilerin isimlerini bulunuz.

```
SELECT _____ isim
FROM musteriler
WHERE _____ yas>10
```

2. En az 2 yolcunun bulunduğu uçuşları bulunuz (ucusNo, ucusTarihi).

```
SELECT f1.ucusNo, f1.ucusTarihi
FROM ucus as f1, ucus as f2
WHERE f1.ucusNo = f2.ucusNo AND f1.ucusTarihi=f2.ucusTarihi AND
_____ f1.no <> f2.no
```

3. En az 5 yolcunun bulunduğu tüm uçuşları ucusNo ve yolcu sayısı olarak bulunuz.

```
SELECT ucusNo, ucusTarihi, count(no) as kacYolcu
FROM ucus
GROUP BY _____ ucusNo, ucusTarihi
HAVING _____ kacYolcu > 4
```

4. "TK102" numaralı uçuşta en az bir kez uçmuş tüm müşterilerin isimlerini bulunuz.

```
SELECT distinct isim
FROM musteri, ucus
WHERE _____ musteri.no=ucus.no AND
_____ ucus.ucusNo="TK102"
```

5. Herhangi bir uçuşta uçmamış müşterilerin isimlerini bulunuz.

```
SELECT distinct isim
FROM musteri _____ left outer join ucus
WHERE _____ ucusNo = NULL
```

6. Hasan isimli müşteri ile aynı uçuşta bulunan tüm müşterilerin isimlerini bulunuz.

Create view HasanUcus (ucusNo) AS

```
SELECT ucusNo
FROM _____musteri natural inner join ucus
WHERE isim = "Hasan"
```



```
SELECT isim
FROM musteri, ucus, HasanUcus
WHERE _____ musteri.no = ucus.no AND
_____ ucus.ucusNo = HasanUcus.ucusNo
```

7. “10/08/2012” tarihinde “TK101” no’lu uçuşta bulunan yolcuların sayısını bulunuz.

```
SELECT _____ count(no )  
  
FROM ucus  
  
WHERE ucusNo= “TK101” AND ucusTarihi=“1/2/1999”
```

8. En fazla yolcunun uçtuğu şehri bulunuz.

Create view sehir_yolcu (nereye, KacKisi) AS

```
SELECT nereye, count(*)  
  
FROM ucus natural inner join ucusBilgi  
  
GROUP BY nereye
```

Create view enfazlaYolcu(KacKisi) AS

```
SELECT _____ max(KacKisi)  
  
FROM _____ sehir_yolcu  
  
SELECT nereye  
  
FROM _____ sehir_yolcu, enfazlaYolcu  
  
WHERE _____ sehir_yolcu.KacKisi =  
  
enfazlaYolcu.KacKisi
```

9. İstanbul’a toplam kaç kişi uçtuğunu bulunuz (eger bir kişi birden fazla uçmuşsa, uçuşlarından sadece bir tanesi sayılacak)

```
SELECT _____ count (distinct no)  
  
FROM ucus natural inner join ucusBilgi  
  
WHERE nereye = “İstanbul”
```

