

Online bilgi yarışması yapan bir web sitesi için veritabanı hazırlanacaktır. Kullanıcıların il bilgilerinin de dikkate alınarak üyelik aşamasından sonra yarışmaya katılabildiği sitede, sorular üyeler tarafından dört şıklı olarak zorluk, puan ve kategori(tarih, spor, kültür vb.) kriterlerine göre gönderilmektedir ve 3 kişilik bilim kurulu tarafından onaylandıktan sonra veritabanına eklenmektedir. Bir üyenin günde en fazla 3 kere yarışabildiği sitede, üyelerin alınan puan, kaç kere yarıştığı, kaç soru gönderdiği gibi yarışma istatistikleri de tutulmaktadır. Yarışacak üye bir kategori seçtikten sonra kendisine kolaydan zora doğru rastgele 20 soru sorulmaktadır ve aynı zorluktaki soruların puanları eşittir. Bunun yanında yarışmada joker olarak yeni soru, arkadaşla sorma ve üyelere sorma seçenekleri de vardır. Ayrıca site üzerinde üyeler kendi arkadaş listesinden seçim yaparak aralarında da bilgi yarışması yapabilmektedir. Aylık olarak en yüksek puana sahip 3 üyeye de çeşitli hediyeler verilmektedir. Buna göre:

1. Verilen açıklamalara göre veri tabanını, veritabanının tablolarını, tablodaki alanları, her bir alanın veri tipini, birincil ve ikincil anahtarlarını veritabanı kurallarına göre tasarlayınız ve verilen açıklamalara göre bu veritabanındaki tüm tabloları SQL kodları ile SQL Server programında oluşturunuz ve veritabanını kodlarken kısıtlayıcılardan her birisini en az bir kere kullanınız. Veritabanını en az tablo ile tüm açıklamaları karşılayacak şekilde tasarlayınız. Veritabanının ilişkisel diyagramını SQL ile çıkarınız.

```
create database bilgiyarismasi1
use bilgiyarismasi1
create table uye(
uye_id int not null identity(1,1) primary key,
uye_adi varchar(30),
uye_soyadi varchar(50),
il varchar(30),
e_posta varchar(100),
parola varchar(10)
)

create table hediye(
hediye_id int not null identity(1,1) primary key,
hediye_adi varchar(50),
uye_id int foreign key references uye(uye_id)
)

create table uye_arkadas(
uye_arkadas_id int not null primary key,
uye_id int foreign key references uye(uye_id),
arkadas_id int foreign key references uye(uye_id),
tarih date
)

create table gelen_soru_havuzu_bk(
gelen_id int not null primary key,
soru varchar(200),
a varchar(50),
b varchar(50),
c varchar(50),
d varchar(50),
cevap varchar(10),
uye_id int foreign key references uye(uye_id),
```

```

kisi1_onay varchar(50),
kisi2_onay varchar(50),
kisi3_onay varchar(50))

create table kategori(
kategori_id int not null primary key,
kategori_ad varchar(50)
)

create table soru(
soru_id int not null primary key,
soru varchar(200),
a varchar(50),
b varchar(50),
c varchar(50),
d varchar(50),
cevap varchar(10),
zorluk varchar(50),
CONSTRAINT check_zorluk
CHECK (zorluk IN ('kolay', 'orta', 'zor')),
kategori_id int foreign key references kategori(kategori_id),
puan int,
uye_id int foreign key references uye(uye_id)
)

create table yarisma(
yarisma_id int primary key,
yarisma_tarih date,
kategori_id int foreign key references kategori(kategori_id))

create table yarisma_soru(
yarisma_soru_id int NOT NULL UNIQUE,
soru_id int foreign key references soru(soru_id),
yarisma_id int foreign key references yarisma(yarisma_id)
)

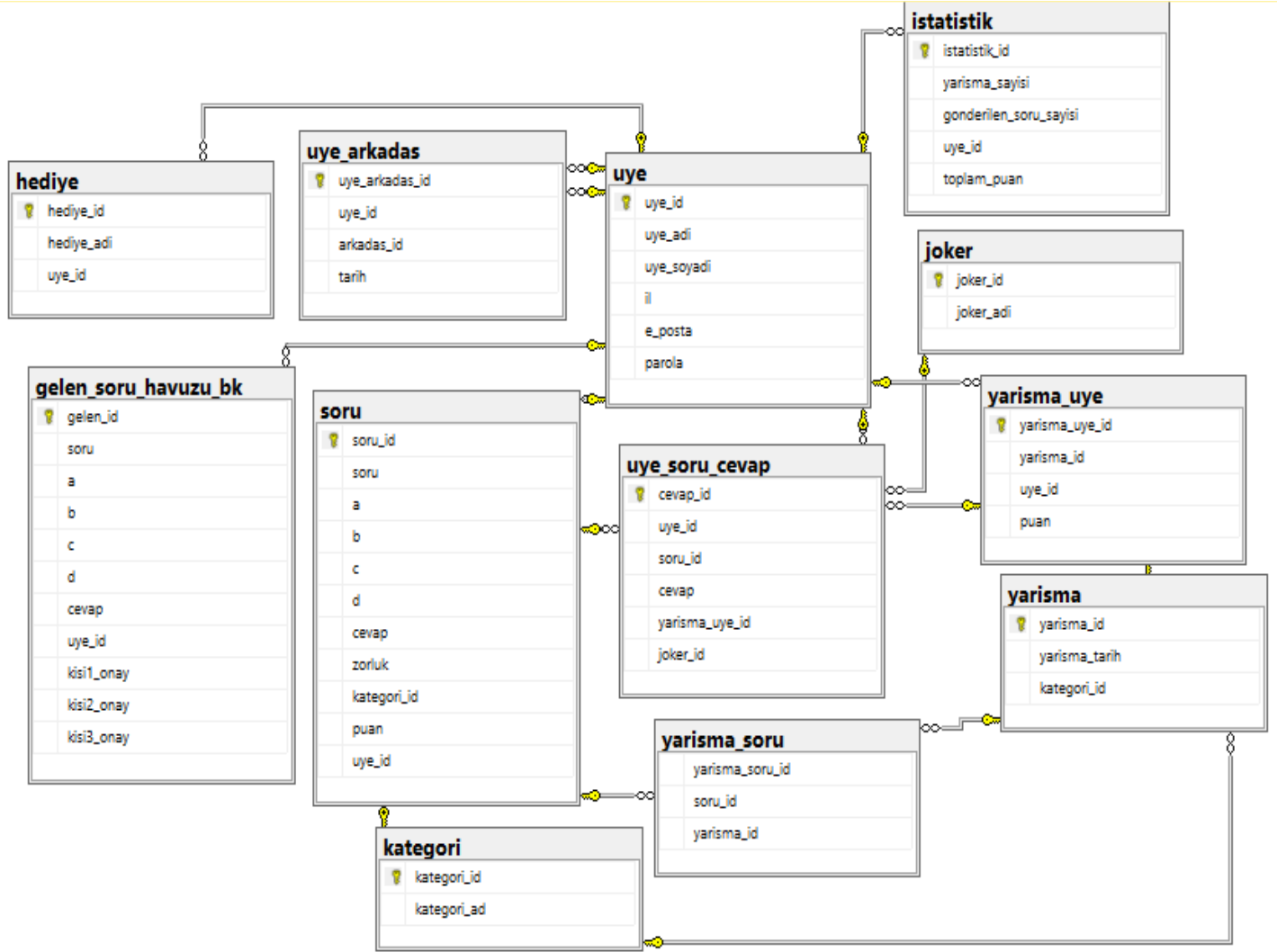
create table joker(
joker_id int primary key,
joker_adi varchar(50))

create table yarisma_uye(
yarisma_uye_id int primary key,
yarisma_id int foreign key references yarisma(yarisma_id),
uye_id int foreign key references uye(uye_id),
puan int)

create table istatistik(
istatistik_id int primary key,
yarisma_sayisi int,
gonderilen_soru_sayisi int,
uye_id int foreign key references uye(uye_id),
toplam_puan int not null check(toplam_puan<=1000000) )

create table uye_soru_cevap(
cevap_id int primary key,
uye_id int foreign key references uye(uye_id),
soru_id int foreign key references soru(soru_id),
cevap varchar(50)
CONSTRAINT check_cevap
CHECK (cevap IN ('dogru', 'yanlis')),
yarisma_uye_id int foreign key references yarisma_uye(yarisma_uye_id),
joker_id int foreign key references joker(joker_id))

```



2. Her bir kategoriden en fazla yarışan üyelerden hangilerinin en yüksek puanı aldığını, hangi kategoride yarıştığını ve kaç soruyu doğru, kaç soruyu yanlış cevapladığını listeleyecek SQL sorgusunu yazınız.

```
select a.uye_id,
(
select count(*) from üye_soru_cevap usc where usc.uye_id=a.uye_id and
usc.yarisma_uye_id=a.kategori_id and cevap ='dogru'
) dogrucevap,
(
select count(*) from üye_soru_cevap usc where usc.uye_id=a.uye_id and
usc.yarisma_uye_id=a.kategori_id and cevap ='yanlis'
) yanliscevap
from (
select y.kategori_id,y.uye_id,max(y.yarismasayisi)as yarismasayisi
from
(
select kategori.kategori_id,uye.uye_id,count(yarisma_uye.yarisma_id)as
yarismasayisi,max(yarisma_uye.puan)as puan
from uye,yarisma_uye,yarisma,kategori
where yarisma.yarisma_id=yarisma_uye.yarisma_id and
yarisma.kategori_id=kategori.kategori_id and uye.uye_id=yarisma_uye.uye_id
group by kategori.kategori_id,uye.uye_id
)y ,uye where y.uye_id=uye.uye_id
)a group by a.kategori_id
```

3. Bugüne kadar hiç soru göndermeyen ancak yarışmaya katılan üyelerin en yüksek puan aldığı yarışmada kaç soruyu doğru cevapladığını, kaç yanlış olduğunu listeleyecek SQL sorgusunu yazınız.

```
select a.uye_adi,a.yarisma_id ,
(select count(*) from üye_soru_cevap usc where usc.uye_id=a.uye_id and
usc.yarisma_uye_id=a.yarisma_id and cevap ='dogru') dogrucevap,
(select count(*) from üye_soru_cevap usc where usc.uye_id=a.uye_id and
usc.yarisma_uye_id=a.yarisma_id and cevap ='yanlış') yanliscevap
from
(select *,
(
SELECT t.yarisma_id

FROM ( SELECT yarisma_id
, MAX(puan) AS max_puan
FROM yarisma_uye
GROUP
BY yarisma_id ) AS m
INNER
JOIN yarisma_uye AS t
ON t.yarisma_id = m.yarisma_id
AND t.puan = m.max_puan where t.uye_id=u.uye_id) yarisma_id

from uye u
where
u.uye_id not in (select distinct uye_id from soru) and u.uye_id in (select distinct
uye_id from yarisma_uye )
```

) a

	uye_adi	yarisma_id	dogrucevap	yanliscevap
1	buket	1	4	1
2	burcu	2	1	0

4. Bugüne kadar 20’den fazla soru gönderen ve hediye alan her bir üyenin her bir kategoriden, her bir zorluk düzeyinde kaç soru gönderdiğini listeleyecek bir SQL sorgusu yazınız ve kullanınız.

```
select kategori.kategori_ad,soru.zorluk,uye.uye_adi,COUNT(soru_id) as "soru sayısı"
from uye,kategori,soru,hediye
where uye.uye_id=soru.uye_id and soru.kategori_id=kategori.kategori_id
and exists(select*from soru where soru.uye_id=uye.uye_id having COUNT(*)>20) and
exists( select*from uye where hediye.hediye_id=uye.uye_id) group by
kategori_ad,zorluk,uye_adi
```

kategori_ad	zorluk	uye_adi	soru sayısı
edebiyat	orta	ayşe	6
fen	kolay	ayşe	5
tarih	zor	ayşe	11