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, arkadaşa 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),

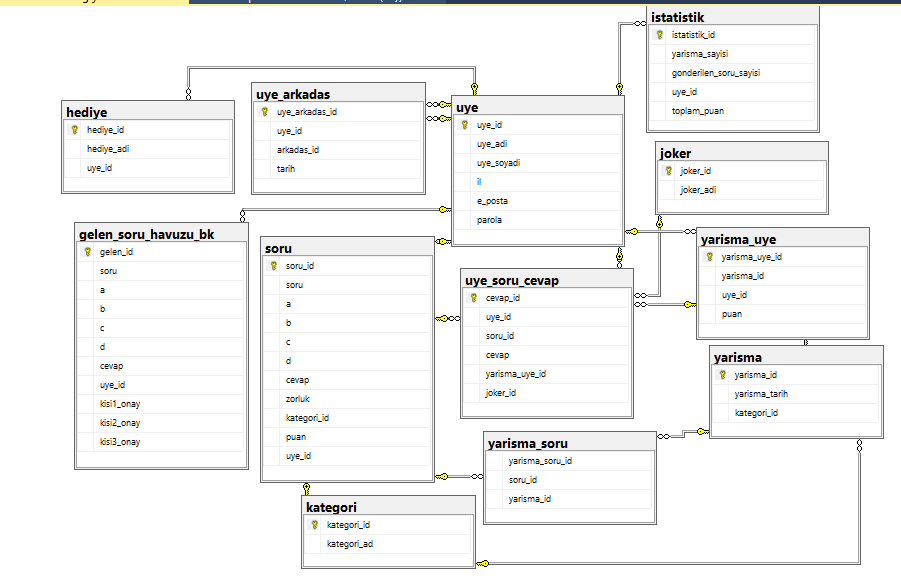
cevap varchar(50)

CONSTRAINT check\_cevap

CHECK (cevap IN ('doğru','yanlış')),

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 yanlışı 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 ='doğru') 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

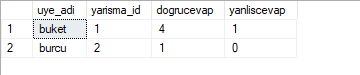
AND t.puan = m.max\_puan where t.uye\_id=u.uye\_id) yarisma\_id

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



**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

