**FUTBOL VERİTABANI**

**Create database FutbolLig;**

**use FutbolLig;**

**Create Table Takim(**

**TakimID int Primary Key identity(1,1),**

**TakimAdi varchar(100),**

**TakimKurYil int,**

**TakimStad varchar(200),**

**TakimTD varchar(200),**

**TakimBaskan varchar(200)**

**);**

**Create Table HakemTur(**

**HakemTurID int Primary Key identity(1,1),**

**HakemTurAdi varchar(50)**

**);**

**Create Table MacTur(**

**MacTurID int primary key identity(1,1),**

**MacTurAdi varchar(100)**

**);**

**Create Table Hakem(**

**HakemID int Primary Key identity(1,1),**

**HakemAdSoyad varchar(200),**

**HakemDTrh datetime,**

**HakemTurID int References HakemTur(HakemTurID)**

**);**

**Create Table Ulke(**

**UlkeID int Primary Key identity(1,1),**

**UlkeAdi varchar(200)**

**);**

**Create Table Futbolcu(**

**FutbolcuID int Primary Key identity(1,1),**

**FurbolcuAd varchar(50),**

**FutbolcuSoyad varchar(100),**

**FutbolcuDTrh datetime,**

**UlkeID int references Ulke(UlkeID),**

**TakimID int references Takim(TakimID)**

**);**

**Create Table TeknikDirektor(**

**TDID int primary key identity(1,1),**

**TDAdSoyad varchar(200),**

**UlkeID int references Ulke(UlkeID)**

**);**

**Create Table TakimLigPuan(**

**TakimID int references Takim(TakimID),**

**Sira int,**

**Puan int**

**);**

**Create Table Mac(**

**MacID int primary key identity(1,1),**

**Takim1ID int references Takim(TakimID),**

**Takim2ID int references Takim(TakimID),**

**MacTrh datetime,**

**HakemID int references Hakem(HakemID),**

**MacTurID int references MacTur(MacTurID),**

**GolTakim1 int,**

**GolTakim2 int**

**);**

**Create Table HakemPuan(**

**HakemID int references Hakem(HakemID),**

**MacID int references Mac(MacID),**

**MacPuan int**

**)**

**Futbol ödev**

**create database ligVeritabani**

**use ligVeritabani**

**create table tblMusabaka(**

**musabakaId int primary key identity(1,1),**

**musabakaAdi nvarchar(100)**

**);**

**create table tblPozisyon(**

**pozisyonId int primary key identity(1,1),**

**pozisyonBilgisi nvarchar(100)**

**);**

**create table tblTakim(**

**takimId int primary key identity(1,1),**

**takimAdi nvarchar(100)**

**);**

**create table tblFutbolcu(**

**futbolcuId int primary key identity(1,1),**

**futbolcuTakimId int foreign key references tblTakim(takimId),**

**futbolcuAdi nvarchar(100),**

**futbolcuFormaNo tinyint,**

**futbolcuPozisyonId int foreign key references tblPozisyon(pozisyonId)**

**);**

**create table tblFikstur(**

**fiksturId int primary key identity(1,1),**

**fiksturMusabakaTarihi date,**

**fiksturMusabakaZamani time(0),**

**fiksturEvSahibiTakimId int foreign key references tblTakim(takimId),**

**fiksturKonukTakimId int foreign key references tblTakim(takimId),**

**fiksturMusabakaId int foreign key references tblMusabaka(musabakaId)**

**);**

**create table tblFutbolcuFikstur(**

**fiksturId int foreign key references tblFikstur(fiksturId),**

**futbolcuId int foreign key references tblFutbolcu(futbolcuId),**

**golSayisi int**

**);**

**Kitap kira**

**create database Kitap\_Kira;**

**use Kitap\_Kira;**

**create table Musteriler(**

**MusteriNo int primary key identity(1,1),**

**MusteriAdi varchar(100),**

**MusteriSoyadi varchar(100),**

**MusteriAdresi varchar(250),**

**MusteriIl varchar(100),**

**MusteriTelefon varchar(15),**

**UyelikTarihi date**

**);**

**create table Kategoriler(**

**KategoriNo int primary key identity(1,1),**

**Aciklamalar varchar(250)**

**);**

**create table Kiralar(**

**FaturaNo int primary key identity(1,1),**

**MusteriNo int references Musteriler(MusteriNo),**

**KiralamaTarihi date,**

**TeslimTarihi date**

**);**

**create table Kitaplar(**

**KitapNo int primary key identity(1,1),**

**KitapAdi varchar(100),**

**KategoriNo int references Kategoriler(KategoriNo),**

**Raiting int,**

**KiralamaFiyatı money,**

**Kiradamı binary**

**);**

**create table KiraDetaylari(**

**ID int primary key identity(1,1),**

**FaturaNo int references Kiralar(FaturaNo),**

**KitapNo int references Kitaplar(KitapNo)**

**);**

**Kütüphane**

**create database Kutuphane**

**use Kutuphane**

**Create table Adresler(**

**Adres\_No int identity(1,1) primary key,**

**Cadde varchar(20),**

**Mahalle varchar(20),**

**Bina\_No varchar(20),**

**Sehir varchar(20),**

**Posta\_Kodu int,**

**Ulke varchar(20)**

**);**

**create table Uyeler(**

**Uye\_No int identity(1,1) primary key,**

**Uye\_Adi varchar(20),**

**Uye\_Soyadi varchar(20),**

**Cinsiyet varchar(5),**

**Telefon varchar(15),**

**E\_Posta varchar(40),**

**Adres\_No int foreign key references Adresler(Adres\_No)**

**);**

**Create Table Kitaplar(**

**ISBN varchar(20),**

**Kitab\_Adi varchar(20),**

**Yayin\_Tarihi varchar(20),**

**S\_Sayisi int,**

**Primary key("ISBN")**

**);**

**Create Table Kutuphane(**

**Kutuphane\_No int primary key identity(1,1),**

**Kutuphane\_Ismi varchar(20),**

**Aciklama varchar(20),**

**Adres\_No int**

**);**

**Create table Emanet(**

**Emanet\_No int identity(1,1) primary key,**

**ISBN varchar(20) foreign key references Kitaplar(ISBN) NOT NULL,**

**Uye\_No int NOT NULL foreign key references Uyeler(Uye\_No),**

**Kutuphane\_No int Not Null foreign key references Kutuphane(Kutuphane\_No),**

**Emanet\_Tarihi Datetime,**

**Teslim\_Tarihi Datetime,**

**);**

**Create Table Kategoriler(**

**Kategori\_No int primary key identity(1,1),**

**Kategori\_Adi varchar(20)**

**);**

**Create Table Yazarlar(**

**Yazar\_No int primary key identity(1,1),**

**Yazar\_Adi varchar(20),**

**Yazar\_Soyadi varchar(20)**

**);**

**Create table Kitap\_Kütüphane(**

**Kutuphane\_No int not null foreign key references Kutuphane(Kutuphane\_No),**

**ISBN varchar(20) Not Null foreign key references Kitaplar(ISBN),**

**Miktar int,**

**Constraint "Kitap\_Kütüphane\_PK" Primary Key("Kutuphane\_No",ISBN)**

**);**

**create table Kitap\_Kategori(**

**ISBN varchar(20) Not NULL foreign key references Kitaplar(ISBN),**

**Kategori\_No int not null foreign key references Kategoriler(Kategori\_No),**

**Constraint "Ktp\_Ktg\_Pk" Primary key("ISBN","Kategori\_No")**

**);**

**Create table Kitap\_Yazar(**

**ISBN varchar(20) Not Null foreign key references Kitaplar(ISBN),**

**Yazar\_No int Not Null foreign key references Yazarlar(Yazar\_No),**

**CONSTRAINT "Kitap\_Yazar\_Pk" Primary key ("ISBN","Yazar\_No")**

**);**

**Okul**

**create database OKUL**

**use OKUL**

**create table Bolum(**

**BNo int identity(1,1) primary key,**

**BAdi varchar(50)**

**);**

**create table Ders(**

**DNo int primary key identity(1,1),**

**DAdi varchar(50),**

**BNo int foreign key references Bolum(BNo)**

**);**

**create table Ogrenci(**

**ONo bigint primary key,**

**Adi varchar(50),**

**soyadi varchar(50),**

**BNo int foreign key references Bolum(BNo)**

**);**

**create table Puan(**

**ONo bigint foreign key references Ogrenci(ONo),**

**BNo int foreign key references Bolum(BNo),**

**Vize int,**

**Final int**

**);**

**Tiyatro veritabanı örnek**

**create database Tiyatro**

**use Tiyatro**

**create table Adres(**

**IlPlaka int primary key,**

**IlAdi varchar(25)**

**)**

**create table Salon(**

**SalonId int primary key identity(1,1),**

**SalonAdi varchar(50)**

**);**

**Create table UyeTipi(**

**TipId int primary key identity(1,1),**

**TipAdi varchar(50),**

**BiletTutari money**

**);**

**Create table Egitim(**

**EgitimId int primary key identity(1,1),**

**EgitimAdi varchar(50),**

**);**

**create table Oyunlar(**

**OyunId int primary key identity(1,1),**

**OyunAdi varchar(50),**

**SalonId int foreign key references Salon(SalonID),**

**OyunTarihi date,**

**OyunSaat time**

**);**

**Create table Uye(**

**UyeId int primary key identity(1,1),**

**UyeAdi varchar(30),**

**UyeSoyadi varchar(30),**

**UyeTipi int foreign key references UyeTipi(TipId),**

**Adres int foreign key references Adres(IlPlaka),**

**);**

**Create table UyeFaaliyet(**

**FaliyetId int primary key identity(1,1),**

**UyeId int foreign key references Uye(UyeID),**

**Oyun int foreign key references Oyunlar(OyunId),**

**Egitim int foreign key references Egitim(EgitimId)**

**);**

***SORULAR***

Yeni kurulmakta olan bir GSM şirketinin ihtiyaçlarını karşılamak üzere yazılacak program için bir veritabanı tasarlanacaktır. Şirketin genel yapısı ve yapılacak işler şu şekilde açıklanmıştır.

Şirket ilk olarak bölge bayilikleri kuracaktır. Bu bölge bayiliklerine bağlı olarakta alt bayilikler verilecektir. Hem bölge bayiliği hem de buna bağlı alt bayilikler için kişilerden adı ve soyadı, adres, il, telefon, faks ve firma ismi bilgileri alınacaktır. Bu adımdan sonra şirket müşterilerini kabul etmeye başlayacaktır.

Abonelik sistemi için tarifeler belirlenmiş olup abone olmak isteyen kişiler istediği tarifeyi seçebilmektedir. Abonelerin ise; telefon numarası, adı ve soyadı, ev adresi, iş adresi, ili, nüfus cüzdan seri no, seçtiği tarife, aboneliğini yurtdışına açtırmak isteyip istemediği, aboneliğini hangi bayilikten ve hangi tarihte açtırdığı bilgileri alınmaktadır. Tarife bilgileri ise tarife adı, dakika konuşma ücreti ve SMS ücreti alanlarından oluşmaktadır.

Ayrıca abonelerin web üzerinden işlem yapabilmeleri için kullanıcı adı, şifre bilgisi, en son giriş yaptığı tarih ve bir ayda kaç kez giriş yaptığı da tutulmaktadır.

Bundan sonra kişilerin her ay sonunda o ay sonunda yaptıkları konuşmalara ait bilgileri içeren bir fatura düzenlenmektedir. Bu faturada kişinin o ay kaç dakika konuştuğu, kaç mesaj attığı, bu ücretlere karşılık gelen toplam borcu ve son ödeme tarihi bilgileri yer almaktadır. Hesaplamalar yapılırken kişilerin abone oldukları tarifelere göre ücretler hesaplanmaktadır. Kişiler faturalarını zamanında ödemedikleri takdirde %8 gecikme faizi uygulanmaktadır.

Hazırlanacak olan bu veritabanına göre aşağıdaki soruları cevaplandırınız.

1. Verilen açıklamalara göre veri tabanını, veritabanının tablolarını, tablodaki alanları, her bir alanın veri tipini, birincil ve ikincil anahtarlarını veritabanı tasarlama kurallarına göre tabloları çizerek tasarlayınız?
2. Verilen açıklamalara göre veri tabanını, veritabanının tablolarını, tablodaki alanları, her bir alanın veri tipini, birincil ve ikincil anahtarlarını veritabanı tasarlama kurallarına göre DDL kodları kullanarak kodlayınız?
3. Bu veritabanının varlık-ilişki modelini çiziniz? Her bir tabloda ne tür ilişkiler olduğunu gösteriniz?
4. Toplam abone sayısı 100’ den büyük olup firma ismi 3 harfli olan ve son harfi Y ile biten bölge bayiliğine bağlı olan alt şubeleri firma isimlerine göre A-Z arasında sıralayacak şekilde listeleyen sorguyu yazınız?
5. Veritabanında her bir bölge bayiliğine kayıtlı kaçar tane alt bölge bayisi ve her alt bölge bayiliğine bağlı kaçar tane abone olduğunu listeleyen sorguyu yazınız?
6. Mevcut ay içerisinde en fazla süre ile konuşma yapan abonenin bilgilerini ve hangi tarifeyi kullandığı bilgilerini listeleyen sorguyu yazınız?
7. Veritabanına bilgilerini kendiniz belirleyeceğiniz şekilde yeni bir alt bayii ekleyen sorguyu yazınız. Daha önce başka tablolara kayıt girilmesi gerekiyorsa onları da ekleyerek başlayınız?
8. Bilgilerini kendiniz belirleyeceğiniz bir kullanıcının veritabanı üzerinde sadece kayıt listeleme yapabileceğini; ama ekleme, silme ve güncelleme yapamayacağını belirleyecek soruları yazınız?
9. Dakikası 0.5 TL, SMS ücreti 0.05 TL olan ZOOM isimli tarifeyi kullanan müşterilerin bilgilerini ve gecikme zammı olmadan müşterilerin ne kadar fatura ödeyeceğini hesaplayacak sorguyu yazınız?
10. Son bir ayda web üzerinden en çok giriş yapmış müşterinin bilgilerini ve hangi tarifeyi kullandığını listeleyecek sorguyu yazınız?

***CEVAPLAR***

|  |
| --- |
| 1. Veritabanı tasarımı |
| 1. Veritabanı oluşturma  |  |  | | --- | --- | | Create Database GSM  use GSM  --Soru2  Create Table il(  ilplaka int Primary Key,  ilAdi varchar(100)  )  Create Table BolgeBayii(  BBNo int Primary Key identity(1,1),  BBAdi varchar(50),  BBAdres varchar(200),  BBTel varchar(13),  BBFaks varchar(13),  BBFirmaAdi varchar(100),  ilPlaka int Foreign Key References il(ilPlaka),  YetkiliAdSoyad varchar(100)  )  Create Table AltBayii(  ABNo int Primary Key identity(1,1),  ABAdi varchar(50),  ABAdres varchar(200),  ABTel varchar(13),  ABFaks varchar(13),  ABFirmaAdi varchar(100),  ilPlaka int Foreign Key References il(ilPlaka),  BBNo int Foreign Key References BolgeBayii(BBNo),  YetkiliAdSoyad varchar(100)  )  Create Table Tarife(  TNo int Primary Key identity(1,1),  TAdi varchar(100),  DakikaUcreti money,  SMSUcreti money  ) | Create Table Abone(  TelNo varchar(15) Primary Key,  TCNo bigint Not Null,  AAdi varchar(50),  ASoyadi varchar(50),  AEvAdres varchar(250),  AisAdres varchar(250),  NC\_SeriNo varchar(10),  AYutDisi bit,  AKTarihi datetime,  TNo int Foreign Key References Tarife(TNo),  ilPlaka int Foreign Key References il(ilPlaka),  ABBNo int Foreign Key References AltBayii(ABNo)  )  Create Table Fatura(  FNo int Primary Key identity(1,1),  ToplamUcret money,  ToplamKonusma int,  ToplamSMS int,  SonTarih datetime,  TelNo varchar(13) Foreign Key References Abone(TelNo)  )  Create Table WebGiris(  TelNo varchar(13) Foreign Key References Abone(TelNo),  Sifre varchar(8),  SonGTrh date,  GirisSayisi int  ) |  1. Varlık-ilişki modeli |
| 1. Soru4 sorgusu   --Soru4  select AltBayii.ABAdi,COUNT(Abone.TelNo) AS 'Abone Sayisi' from AltBayii,Abone where AltBayii.ABNo=Abone.ABNo and AltBayii.ABAdi LIKE '\_\_Y' group by AltBayii.ABAdi having(COUNT(Abone.TelNo)>=100)order by AltBayii.ABAdi desc   1. Soru5 sorgusu   --Soru5  select BolgeBayii.BBAdi,AltBayii.ABAdi,COUNT(Abone.TelNo) AS 'Abone Sayisi' from BolgeBayii,AltBayii,Abone where Abone.ABNo=AltBayii.ABNo and AltBayii.BBNo=BolgeBayii.BBNo group by BolgeBayii.BBAdi,AltBayii.ABAdi   1. Soru6 sorgusu   --Soru6  select Abone.TelNo,Abone.AAdi,Abone.ASoyadi,Tarife.TAdi,Fatura.ToplamKonusma from Tarife inner join Abone on Tarife.TNo=Abone.TNo inner join Fatura on Fatura.TelNo=Abone.TelNo and Fatura.ToplamKonusma=(select MAX(Fatura.ToplamKonusma) from Fatura)   1. Soru7 sorgusu   --Soru7  select \* from AltBayii  insert into AltBayii(ABAdi,ABAdres,ABTel,ABFaks,ABFirmaAdi,ilPlaka,BBNo,YetkiliAdSoyad) values ('A21','Bilecik','12345','123450','A21',11,1,'Kamil Yanar')   1. Soru8 sorgusu   --Soru 8  grant select to user1 on GSM  deny update,delete,insert to user1 on GSM   1. Soru9 sorgusu   --Soru9  select Abone.TelNo,Abone.AAdi,Abone.ASoyadi,Tarife.TAdi,(Tarife.DakikaUcreti\*Fatura.ToplamKonusma+Tarife.SMSUcreti\*Fatura.ToplamSMS) AS 'Fatura Tutarý' from Abone,Tarife,Fatura where Abone.TNo=Tarife.TNo and Abone.TelNo=Fatura.TelNo and Tarife.TAdi='ZOOM'   1. Soru 10 sorgusu   --Soru10  select Abone.TelNo,Abone.AAdi,Abone.ASoyadi,Tarife.TAdi,WebGiris.GirisSayisi from Abone,Tarife,WebGiris where Abone.TNo=Tarife.TNo and WebGiris.TelNo=Abone.TelNo and WebGiris.GirisSayisi=(select MAX(WebGiris.GirisSayisi) from WebGiris) |

2015-2016 Vize Sınavı

FUTBOL LİG

Create Database Futbol\_Lig

Use Futbol\_Lig

Create Table Ulke(

Ulke\_Id int Primary Key identity(1,1),

Ulke\_Adi varchar(200)

)

Create Table Teknik\_Direktor(

Teknik\_Direktor\_Id int Primary Key identity(1,1),

Teknik\_Direktor\_Ad\_Soyad varchar(200),

Ulke\_Id int Foreign Key References Ulke(Ulke\_Id)

)

Create Table Takim(

Takim\_Id int Primary Key identity(1,1),

Takim\_Adi varchar(100),

Takim\_Kur\_Yil int,

Takim\_Stad varchar(200),

Takim\_Teknik\_Direktor\_Id int Foreign Key References Teknik\_Direktor(Teknik\_Direktor\_Id),

Takim\_Baskan varchar(200)

)

Create Table Hakem\_Tur(

Hakem\_Tur\_Id int Primary Key identity(1,1),

Hakem\_Tur\_Adi varchar(50)

)

Create Table Mac\_Tur(

Mac\_Tur\_Id int Primary Key identity(1,1),

Mac\_Tur\_Adi varchar(100)

)

Create Table Hakem(

Hakem\_Id int Primary Key identity(1,1),

Hakem\_Ad\_Soyad varchar(200),

Hakem\_Dogum\_Tarih datetime,

Hakem\_Tur\_Id int Foreign Key References Hakem\_Tur(Hakem\_Tur\_Id)

)

Create Table Futbolcu(

Futbolcu\_Id int Primary Key identity(1,1),

Futbolcu\_Ad varchar(100),

Futbolcu\_Soyad varchar(100),

Futbolcu\_D\_Tarih datetime,

Ulke\_Id int Foreign Key References Ulke(Ulke\_Id),

Takim\_Id int Foreign Key References Takim(Takim\_Id)

)

Create Table Takim\_Lig\_Puan(

Takim\_Id int Foreign Key References Takim(Takim\_Id),

Sira int,

Puan int

)

Create Table Mac(

Mac\_Id int Primary Key identity(1,1),

Takim1\_Id int Foreign Key References Takim(Takim\_Id),

Takim2\_Id int Foreign Key References Takim(Takim\_Id),

Mac\_Tarihi datetime,

Hakem\_Id int Foreign Key References Hakem(HAkem\_Id),

Mac\_Tur int Foreign Key References Mac\_Tur(Mac\_Tur\_Id),

Gol\_Takim1 int,

Gol\_Takim2 int,

)

Create Table Hakem\_Puan(

Hakem\_Id int Foreign Key References Hakem(Hakem\_Id),

Mac\_Id int Foreign Key References Mac(Mac\_Id),

Mac\_Puan int

)

DATABASE DİAGRAM



🡪 3 Tüm takımları şu anki durumda sıralarına göre; takım adı, galibiyet sayısı, beraberlik sayısı, attığı gol sayısı, yediği gol sayısı, averaj bilgisi ve puanını gösterecek şekilde sıralayacak SQL sorgusunu yazınız.

SELECT Takim.Takim\_Id,Takim\_Lig\_Puan.Sira,Takim\_Lig\_Puan.Puan,Takim.Takim\_Adi,AtGol,YeGol, AtGol-YeGol AS Average,Maglubiyet,galib As Galibiyet, berabere from Takim,Takim\_Lig\_Puan,(select tk1,gol1+gol2 as AtGol from (select Takim.Takim\_Id,Takim.Takim\_Adi AS tk1, SUM(Mac.Gol\_Takim1) As gol1 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id group by Takim.Takim\_Id,Takim.Takim\_Adi) AS dervied, (select Takim.Takim\_Id,Takim.Takim\_Adi tk2, SUM(Mac.Gol\_Takim2) AS gol2 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id GROUP BY Takim.Takim\_Id,Takim.Takim\_Adi) der2 where tk1=tk2) as den,(select tk2, gol1+gol2 as YeGol from (select Takim.Takim\_Id,Takim.Takim\_Adi AS tk1,SUM(Mac.Gol\_Takim2) As gol1 from Takim, Mac where Takim.Takim\_Id = Mac.Takim1\_Id group by Takim.Takim\_Id,Takim.Takim\_Adi) As dervied,(select Takim.Takim\_Id, Takim.Takim\_Adi tk2, SUM(Mac.Gol\_Takim1) as gol2 from Takim,Mac where Takim.Takim\_Id= Mac.Takim2\_Id group by Takim.Takim\_Id,Takim.Takim\_Adi) AS der2 where tk1=tk2) as dene, (select tk3, g1+g2 as galibiyet from (select Takim.Takim\_Id,Takim.Takim\_Adi as tk3,COUNT(Mac.Takim1\_Id) as g1 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1>Mac.Gol\_Takim2 group by takim.Takim\_Id,Takim.Takim\_Adi) as dervied, (select Takim.Takim\_Id,Takim.Takim\_Adi tk4, COUNT(Mac.Takim2\_Id) As g2 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1< Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi) as der2 where tk3=tk4) as gal, (select tk1 Tadi, Sum(g1) as maglubiyet from(select Takim.Takim\_Id,Takim.Takim\_Adi as tk1, COUNT(Mac.Takim1\_Id) as g1 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1< Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

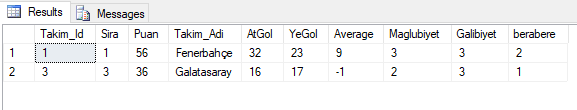
select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g2 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1>Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi) as d group by tk1) as mag,(select tkg as tkadi, SUM(g3) as galib from (select Takim.Takim\_Id,Takim.Takim\_Adi as tkg, COUNT(Mac.Takim1\_Id) AS g3 from Takim,Mac where takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1>mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g4 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1<Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi) gal group by tkg) as glb,(select tkb Takadi, SUM(gb) as berabere from (select Takim.Takim\_Id,Takim.Takim\_Adi as tkb, COUNT(Mac.Takim1\_Id) AS gb from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1=Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

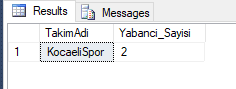
select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g7 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1=Mac.Gol\_Takim2 group by Takim.Takim\_Id,Takim.Takim\_Adi) as d group by tkb) as brb where Takim.Takim\_Id=Takim\_Lig\_Puan.Takim\_Id and tk1=Takim.Takim\_Adi and tk2=tk1 and tk3=tk1 and Tadi=tk1 and tkadi=tk1 and Takadi=tk1 order by Puan desc



🡪 Bu futbol ligi veritabanının Türkiye için tasarlandığını varsayarsak; yabancı teknik direktörlerin çalıştırdığı takımlar arasında hangi takımda daha fazla yabancı futbolcu bulunmaktadır? Bu sorunun çözümü için bir sql sorgusu yazınız.

select TakimAdi,Yabanci\_Sayisi from (select Takim.Takim\_Adi as TakimAdi,Count (Futbolcu.Futbolcu\_Id) as Yabanci\_Sayisi from Takim,Futbolcu where Takim.Takim\_Id=Futbolcu.Takim\_Id

and Takim.Takim\_Teknik\_Direktor\_Id in (select Teknik\_Direktor.Teknik\_Direktor\_Id from Teknik\_Direktor where Teknik\_Direktor.Ulke\_Id in (select Ulke.Ulke\_Id from Ulke where Ulke.Ulke\_Adi<>'Turkiye')) and Futbolcu.Ulke\_Id in(select Ulke.Ulke\_Id from Ulke where Ulke.Ulke\_Adi<>'Turkiye') group by Takim.Takim\_Adi) takim where Takim.Yabanci\_Sayisi= (select MAX(Yabanci\_Sayisi) from (select Takim.Takim\_Adi as TakimAdi,COUNT(Futbolcu.Futbolcu\_Id)as Yabanci\_Sayisi from Takim,Futbolcu where Takim.Takim\_Id=Futbolcu.Takim\_Id and Takim.Takim\_Teknik\_Direktor\_Id in (select Teknik\_Direktor.Teknik\_Direktor\_Id from Teknik\_Direktor where Teknik\_Direktor.Ulke\_Id in (select Ulke.Ulke\_Id from Ulke Where Ulke.Ulke\_Adi<>'Turkiye')) and Futbolcu.Ulke\_Id in (select Ulke.Ulke\_Id from Ulke where Ulke.Ulke\_Adi<>'Turkiye') group by Takim.Takim\_Adi) takim2)



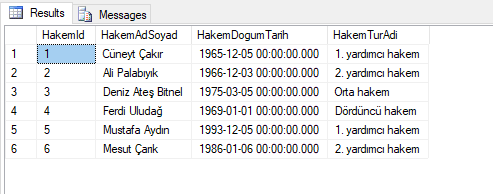
-------------------------------------------------------------------------------------------------------------------------------------------

* 5. A isimli takımın maçlarını yönetmeyen hakemlerin bilgilerini ekrana listeleyecek bir view yazınız ve kullanınız.

Create View Mac\_Yonetmeyen\_Hakem(HakemId,HakemAdSoyad, HakemDogumTarih,HakemTurAdi) AS

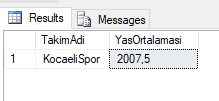
select Hakem.Hakem\_Id,Hakem.Hakem\_Ad\_Soyad, Hakem.Hakem\_Dogum\_Tarih, Hakem\_Tur.Hakem\_Tur\_Adi from Hakem,Hakem\_Tur where Hakem.Hakem\_Tur\_Id=Hakem\_Tur.Hakem\_Tur\_Id and Hakem.Hakem\_Id not in (select Mac.Hakem\_Id from Mac where Mac.Takim1\_Id in (select Takim.Takim\_Id from Takim where Takim.Takim\_Adi='Fenerbahce')) and Hakem.Hakem\_Id not in (select Mac.Hakem\_Id from Mac where Mac.Takim2\_Id in (select Takim.Takim\_Id from Takim where Takim.Takim\_Adi='Fenerbahce'))

select \* from Mac\_Yonetmeyen\_Hakem



* 6 Takımın Yaş ortalamsı en düşük takım hangisidir? Bu cevabı verecek SQL sorgusunu yazınız..

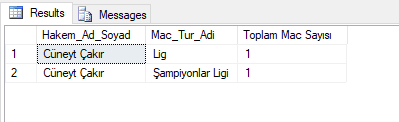
select TakimAdi,YasOrtalamasi from (select TakimAdi, MAX(TakimYasOrtalamasi) AS YasOrtalamasi from (select Takim.Takim\_Adi as TakimAdi,AVG(CAST(SUBSTRING(CAST(Futbolcu.Futbolcu\_D\_Tarih as varchar(50)),8,4) as real)) AS TakimYasOrtalamasi from Takim, Futbolcu where Takim.Takim\_Id=Futbolcu.Takim\_Id group by Takim.Takim\_Adi) as minyas group by TakimAdi) minyas2 where YasOrtalamasi=(select MAX (TakimYasOrtalamasi) from (select Takim.Takim\_Adi as TakimAdi, AVG(CAST(SUBSTRING(CAST(Futbolcu.Futbolcu\_D\_Tarih as varchar(50)),8,4) as real)) AS TakimYasOrtalamasi from Takim,Futbolcu where Takim.Takim\_Id=Futbolcu.Takim\_Id group by Takim.Takim\_Adi) as yas)



-------------------------------------------------------------------------------------------------------------------------------------------

--> EN fazla puan toplayan hakemin her bir maç türünden kaç tane maç yönettiğini gösterecek bir sql sorgusu yazınız.

select Hakem.Hakem\_Ad\_Soyad,Mac\_Tur.Mac\_Tur\_Adi,COUNT(Mac.Mac\_ID) AS 'Toplam Mac Sayısı' from Hakem,Mac,Mac\_Tur,Hakem\_Puan where Mac.Mac\_Tur=Mac\_Tur.Mac\_Tur\_Id and Hakem.Hakem\_Id=Mac.Hakem\_Id and Hakem.Hakem\_Id=Hakem\_Puan.Hakem\_Id and Hakem\_Puan.Mac\_Id=Mac.Mac\_Id and Hakem.Hakem\_Ad\_Soyad in (select Hakem from (select Hakem.Hakem\_Ad\_Soyad as Hakem,Sum(Hakem\_Puan.Mac\_Puan) AS hakempuan from Hakem,Hakem\_Puan where Hakem.Hakem\_Id=Hakem\_Puan.Hakem\_Id group by Hakem.Hakem\_Ad\_Soyad) AS hakemmac where hakempuan= (select MAX (hakempuan) from(select SUM(Hakem\_Puan.Mac\_Puan) AS hakempuan from Hakem,Hakem\_Puan where Hakem.Hakem\_Id=Hakem\_Puan.Hakem\_Id group by Hakem.Hakem\_Ad\_Soyad) hakemmac2)) group by Hakem.Hakem\_Ad\_Soyad,Mac\_Tur.Mac\_Tur\_Adi



-------------------------------------------------------------------------------------------------------------------------------------------

--> Bir maçın sonucuna göre; galibiyet, mağlubiyet ve beraberlik durumlarında puan cetvelinde ilgili takımların puan ve sıralamalarında gerekli güncellemeleri yapacak bir SQL sorgusunu yazınız.

update Takim\_Lig\_Puan set Puan=Puan+3 where Takim\_Id in(select Takim\_Id from Takim where Takim\_Id in (select Takim1\_Id from Mac where Mac.Gol\_Takim1>Mac.Gol\_Takim2 and Mac.Mac\_Id=10))

update Takim\_Lig\_Puan set Puan=Puan+3 where Takim\_Id in(select Takim\_Id from Takim where Takim\_Id in (select Takim2\_Id from Mac where Mac.Gol\_Takim1<Mac.Gol\_Takim2 and Mac.Mac\_Id=10))

update Takim\_Lig\_Puan set Puan=Puan+1 where Takim\_Id in(select Takim\_Id from Takim where Takim\_Id in ((select Takim1\_Id from Mac where Mac.Gol\_Takim1=Mac.Gol\_Takim2 and Mac.Mac\_Id=10),(select Takim2\_Id from Mac where Mac.Gol\_Takim1=Mac.Gol\_Takim2 and Mac.Mac\_Id=10)))

9. Hem futbolcular için hem de teknik direktörler için daha önceki takım bilgilerini veritabanında tutmak için bu veritabanı yeterli midir? Yeterli ise nasıl ulaşırız, değilse öneriniz nedir?

Bu veritabanında, hem futbolcular için hem de teknik direktörler için daha önceki takım bilgilerini tutmak için bu veritabanı yeterli değildir. Görüleceği üzere hem teknik direktörler için hem de futbolcular için sadece bir takım bilgisine ulaşılabilmektedir. Yani en son takım bilgilerine ulaşılabilmektedir. Bu şekilde kayıtlar tutulmak istenirse FutbolcuTakım ve TeknikDirektorTakım isimli iki yeni tablo oluşturulup, futbolcular ve teknik direktörler takım değiştirdiklerinde yeni ve eski bilgiler oluşturulacak ikincil anahtarlar aracılığıyla bu tablolarda tutulabilir.

--> Kendi ligi dışında uluslararası maç yapan takımların bilgilerini ve bu takımların her birisinin bu maçlardaki galibiyet sayısını, mağlubiyet sayısını ve beraberlik sayısını listeleyecek bir sql sorgusu yazınız. (bu maçların bu veritabanındaki puan tablosuna etkisi olmadığı varsayılacaktır.)

select Takim.Takim\_Adi,galibiyet,maglubiyet,berabere from Takim,Mac,(select tkg as tkadi, SUM(g3) as galibiyet from (select Takim.Takim\_Id,Takim.Takim\_Adi as tkg,COUNT(Mac.Takim1\_Id) AS g3 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1>Mac.Gol\_Takim2 and Mac.Mac\_Tur=3 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g4 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1<Mac.Gol\_Takim2 and Mac.Mac\_Tur =3 group by Takim.Takim\_Id,Takim.Takim\_Adi) gal group by tkg) as galib,(select tkm as tadi, SUM(g3) as maglubiyet from (select Takim.Takim\_Id,Takim.Takim\_Adi as tkm, COUNT(Mac.Takim1\_Id) AS g3 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1<Mac.Gol\_Takim2 and Mac.Mac\_Tur=3 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g4 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1>Mac.Gol\_Takim2 and Mac.Mac\_Tur =3 group by Takim.Takim\_Id,Takim.Takim\_Adi) mag group by tkm) as maglup,(select tkm as tkmadi, SUM(g3) as berabere from (select Takim.Takim\_Id,Takim.Takim\_Adi as tkm, COUNT(Mac.Takim1\_Id) AS g3 from Takim,Mac where Takim.Takim\_Id=Mac.Takim1\_Id and Mac.Gol\_Takim1=Mac.Gol\_Takim2 and Mac.Mac\_Tur=3 group by Takim.Takim\_Id,Takim.Takim\_Adi

union

Select Takim.Takim\_Id,Takim.Takim\_Adi tk2, COUNT(Mac.Takim2\_Id) AS g4 from Takim,Mac where Takim.Takim\_Id=Mac.Takim2\_Id and Mac.Gol\_Takim1=Mac.Gol\_Takim2 and Mac.Mac\_Tur=3 group by Takim.Takim\_Id,Takim.Takim\_Adi) ber group by tkm) as beraber where Takim.Takim\_Adi=tkadi and Mac.Mac\_Tur=3

**“Bir grup insanın bir araya gelerek enstrümantal ya da vokal olarak gerçekleştirdikleri müzik türüne grup ya da BAND denilmektedir.**

**Bir kişi BAND olarak müzik yapanlar için albüm koleksiyonu ile ilgilenmektedir. Aynı zamanda, bu kişi derleme albümleri koleksiyonuna dahil etmemektedir, yani her albümdeki şarkılar, sadece o albüm içindir ve sadece ilgili BANDe aittir. Her BANDe ait albüm için bu kişi albümün başlık\ad,yıl ve BANDin kaçıncı albümü olduğunu veritabanında tutmak istemektedir. Ayrıca, bu kişi için şarkıların adı, uzunluğu, track numaralarısı,sözleri ve söz yazarı bilgileri önemlidir. Tabii ki, iki ayrı BAND aynı şarkıyı kaydederse, track numaraları ve uzunlukları farklı olabilir. Bunun yanında, her BAND için bu kişi, BANDin her bir üyesinin isim bilgisine ve BANDin ismine ihtiyaç duymaktadır. Her bir BAND üyesi için ad, soyad ve ülke bilgisi yeterlidir. Bu koleksiyon için dikkat edilecek nokta ise hem söz yazarları hem de BAND üyeleri birer müzisyendir.**

2. Verilen açıklamalara göre bu veritabanını SQL kodları ile oluşturunuz ve veritabanını kodlarken kısıtlatıcılardan her birisini en az bir kere kullanınız.

Create Database BANDCollection

Use BANDCollection

Create Table Songs(

SongID int primary key identity(1,1),

SongTitle varchar(250) NOT NULL

)

Create Table Countries(

CID int primary key identity(1,1)

Check(CID>=1 and CID<=500),

CName varchar(200)

)

Create Table Bands(

BandID int primary key identity(1,1),

BandName varchar(250) NOT NULL

)

Create Table Albums(

AlbumID int primary key identity(1,1),

AlbumTitle varchar(255) NOT NULL,

AlbumYear int NOT NULL,

BandID int Foreign Key References Bands(BandID),

AlbumNumber smallint

)

Create Table AlbumSongs(

SongID int Foreign Key References Songs(SongID),

AlbumID int Foreign Key References Albums(AlbumID),

TrackNumber int NULL,

TrackLength int NOT NULL,

)

Create Table Musicians(

MusicianID int primary key identity(1,1),

MusicianName varchar(200) NOT NULL,

MusicianSurname varchar(200) NOT NULL,

MusicianCountry int Foreign Key References Countries(CID) DEFAULT(1)

)

Create Table SongWriters(

SongID int Foreign Key References Songs(SongID),

MusicianID int Foreign Key References Musicians(MusicianID)

)

Create Table BandMusicians(

BandID int Foreign Key References Bands(BandID),

MusicianID int Foreign Key References Musicians(MusicianID))

3. ‘BigBAND’ isimli BAND’in bugüne kadar en fazla şarkı koydukları albüm için, bu albüme en fazla şarkı veren söz yazarının ad, soyad ve ülke bilgisini listeleyecek SQL sorgusunu yazınız.

select distinct Musicians.MusicianName,Musicians.MusicianSurname,Countries.CName from

SongWriters,Countries,Musicians where Countries.CID=Musicians.MusicianCountry and

Musicians.MusicianID=SongWriters.MusicianID and SongWriters.MusicianID in (select sw.mid from ( select

SongWriters.MusicianID as mid,Albums.AlbumID as a\_id, COUNT(AlbumSongs.SongID) as s\_id from SongWriters,

AlbumSongs,Albums,Songs where AlbumSongs.SongID=Songs.SongID and Songs.SongID=SongWriters.SongID

and Albums.AlbumID=AlbumSongs.AlbumID group by

SongWriters.MusicianID,Albums.AlbumID) sw where s\_id=(select MAX(s\_id) from (select COUNT(AlbumSongs.SongID) as

s\_id from SongWriters,AlbumSongs,Songs,Albums where AlbumSongs.SongID=Songs.SongID and Songs.SongID=SongWriters.SongID

and Albums.AlbumID=AlbumSongs.AlbumID group by SongWriters.MusicianID,Albums.AlbumID) sw2 ) and a\_id

in

(select bnd.aid from (select Bands.BandName,Albums.AlbumID as aid, COUNT(AlbumSongs.AlbumID) AS

album\_sarki\_sayisi from Bands,Albums,AlbumSongs where Bands.BandID=Albums.BandID and Albums.AlbumID=AlbumSongs.AlbumID and

Bands.BandName = 'BigBAND' group by Bands.BandName,Albums.AlbumID) bnd where bnd.album\_sarki\_sayisi=(

select MAX(album\_sarki\_sayisi) from (select COUNT(AlbumSongs.AlbumID) AS album\_sarki\_sayisi from Bands,Albums,

AlbumSongs where Bands.BandID=Albums.BandID and Albums.AlbumID=AlbumSongs.AlbumID and Bands.BandName='BigBAND' group by

Bands.BandName,Albums.AlbumID)bnd2)))

4. Aynı şarkıyı albümlerine koyan BAND’ler için albüm bilgileri, BAND bilgileri ve şarkı bilgilerini karşılaştırmalı olarak görebileceğimiz bir SQL sorgusu yazınız.

Select Songs.SongTitle,Albums.AlbumTitle,AlbumSongs.TrackNumber,AlbumSongs.TrackLength,Albums.AlbumYear,

Bands.BandName from Bands,Albums,AlbumSongs,Songs where Bands.BandID=Albums.BandID and Albums.AlbumID=AlbumSongs.AlbumID and AlbumSongs.SongID=Songs.SongID

and Songs.SongTitle ='Hello Kity'

5.A isimli BAND’in hiçbir albümünde şarkısı olmayan söz yazarlarının bilgilerini ekrana listeleyecek bir view yazınız ve kullanınız.

Create View SongWriters\_List AS

Select distinct Musicians.MusicianName,Musicians.MusicianSurname,Countries.CName from

Musicians,Countries,SongWriters,Bands,Albums,AlbumSongs,Songs where

Countries.CID=Musicians.MusicianCountry and Musicians.MusicianID=SongWriters.MusicianID and

SongWriters.SongID=Songs.SongID and Songs.SongID=AlbumSongs.SongID and

AlbumSongs.AlbumID=Albums.AlbumID and Albums.BandID=Bands.BandID and Musicians.MusicianID NOT

IN (select SongWriters.MusicianID from SongWriters where SongWriters.SongID IN (select

AlbumSongs.SongID from AlbumSongs where AlbumSongs.AlbumID IN (select Albums.AlbumID from

Albums where Albums.BandID IN (select Bands.BandID from Bands where Bands.BandName='A'))))

select \* from SongWriters\_List

6.En fazla üyeye sahip BAND her bir albümünde kaç şarkı yapmıştır. Bu cevabı verecek SQL sorgusunu yazınız.

Select Bands.BandName,Albums.AlbumTitle,COUNT(AlbumSongs.SongID) AS Sarki\_Sayisi from

Bands,Albums,AlbumSongs,Songs where Bands.BandID=Albums.BandID and

Albums.AlbumID=AlbumSongs.AlbumID and AlbumSongs.SongID=Songs.SongID and Bands.BandID = (select

b\_id from (select Bands.BandID as b\_id,COUNT(BandMusicians.MusicianID) as u\_num from

Bands,BandMusicians where Bands.BandID=BandMusicians.BandID group by Bands.BandID) bmus where

bmus.u\_num=(select MAX(u\_num) from (select COUNT(BandMusicians.MusicianID) as u\_num from

Bands,BandMusicians where Bands.BandID=BandMusicians.BandID group by Bands.BandID) bmus2))

group by Bands.BandName,Albums.AlbumTitle

7. Bugüne kadar her bir söz yazarının, her bir BAND’in, her bir albümünde kaç şarkısı olduğunu gösteren bir SQL sorgusu yazınız.

Select Musicians.MusicianName+' '+Musicians.MusicianSurname 'Song

Writers',Bands.BandName,Albums.AlbumTitle,COUNT(AlbumSongs.SongID) Sarki\_Sayisi from

Musicians,Songs,Albums,AlbumSongs,SongWriters,Bands where

Musicians.MusicianID=SongWriters.MusicianID and SongWriters.SongID=Songs.SongID and

Songs.SongID=AlbumSongs.SongID and AlbumSongs.AlbumID=Albums.AlbumID and

Albums.BandID=Bands.BandID group by Musicians.MusicianName+' '+Musicians.MusicianSurname,

Bands.BandName,Albums.AlbumTitle

8.Tüm albümler içerisinde bulunan ortalama şarkı sayısından daha fazla şarkı bulunan albümlerdeki en fazla uzunluğa sahip şarkının hangi söz yazarına ve hangi BAND(lerin) albümünde bulunduğunu listeleyecek bir SQL sorgusu yazınız.

Select Songs.SongTitle,AlbumSongs.TrackLength,Musicians.MusicianName+' '+Musicians.MusicianSurname

'Song Writers',Bands.BandName,Albums.AlbumTitle from

Songs,SongWriters,Musicians,Bands,AlbumSongs,Albums where

Musicians.MusicianID=SongWriters.MusicianID and SongWriters.SongID=Songs.SongID and

Songs.SongID=AlbumSongs.SongID and AlbumSongs.AlbumID=Albums.AlbumID and

Albums.BandID=Bands.BandID and TrackLength=(Select MAX(TrackLength) from AlbumSongs) and

Albums.AlbumID IN

(Select alb1.a\_id from (select Albums.AlbumID as a\_id,COUNT(AlbumSongs.SongID) ssay from

Albums,AlbumSongs where Albums.AlbumID=AlbumSongs.AlbumID group by Albums.AlbumID) alb1 where

alb1.ssay>(select AVG(alb.ss) from (select Albums.AlbumID as a\_id,COUNT(AlbumSongs.SongID) as

ss from Albums,AlbumSongs where Albums.AlbumID=AlbumSongs.AlbumID group by Albums.AlbumID)

alb))

9. BAND değiştiren bir müzisyenin veritabanında bilgilerinin güncellenmesi esnasında önceki ve sonraki bilgilerini kullanıcıya gösterecek bir OUTPUT yazınız.

declare @musc TABLE(

e\_band varchar(60),

y\_band varchar(60)

)

update BandMusicians set BandID=1

OUTPUT deleted.BandID,inserted.BandID INTO @musc

where MusicianID=4

select \* from @musc

10. En az 9 şarkı içeren albümleri ve BAND isimlerini listeleyecek bir SQL sorgusu yazınız.

Select Albums.AlbumTitle,Bands.BandName,COUNT(AlbumSongs.SongID) Sarki\_Sayisi from

Albums,AlbumSongs,Bands where Albums.AlbumID=AlbumSongs.AlbumID and Albums.BandID=Bands.BandID

group by Albums.AlbumTitle,Bands.BandName having (COUNT(AlbumSongs.SongID)>=9)

11. En eski albüme sahip BAND’in, bu albümü yaptığı yılın asal bir sayı olup olmadığını gösterecek bir T-SQL programı yazınız.

declare @ayil int

select @ayil=MIN(Albums.AlbumYear) from Albums

declare @syc int,@k int

set @k=0

set @syc=1

while(@syc<=@ayil)

Begin

if(@ayil%@syc=0)

Begin

set @k=@k+1

print(CAST(@ayil as varchar(10))+' sayisi '+CAST(@syc as varchar(10))+' sayisina tam bolunur.')

End

set @syc=@syc+1

End

if(@k>=3)

print( CAST(@ayil as varchar(10))+' sayisi asal degildir.')

else

print( CAST(@ayil as varchar(10))+' sayisi asaldir.')

FİLM KİRALAMA FİRMASI

Bir film kiralama firması için program hazırlanacaktır. Bu amaçla yukarıda veri tabanı çizilmiştir. Bu veri tabanında müşterilere ait müşteri numarası, adı, soyadı, cinsiyet, kaç çocuğu olduğu ve eğitim düzeyi bilgileri tutulmaktadır. Firmada satılan filmlere ait film no, film adı, ne tür bir film olduğu ve o an ödünç verilip verilmediği bilgisi tutulmaktadır. Ayrıca müşterilerin daha yakından tanınması ve yeni bir film geldiğinde müşterilere haber verilebilmesi amacıyla firma elinde bulunan filmlerde oynayan oyuncular ve filmlerin yönetmenleri ile ilgili bilgilerde tutmaktadır. Bu amaçla tüm filmlerde ismi geçen oyuncu ve yönetmenlere ait bilgiler oyuncunun tablosunda saklanmaktadır. Tablodaki kişinin bir oyuncu mu yoksa yönetmen mi olduğu bilgisi bu tablodaki oyyon alanındaki bilgi ile öğrenilebilmektedir. Bir filmde oynayan oyuncular ve yönetmene ait bilgiler oyuncufilm tablosunda tutulmaktadır. Bir müşterinin sevdiği oyuncu ve yönetmene ait bilgiler ise musterioyuncu tablosunda saklanmaktadır. Son olarak müşterilerin hangi filmleri ödünç aldıkları, hangi tarihde aldıkları, hangi tarihte getirecekleri ve getirip getirmediklerine ait bilgiler

de musterifilm tablosunda saklanmaktadır.

Aşağıdaki soruları yukarıdaki veri tabanına göre cevaplayınız.

1. Tasarlanmış olan bu veri tabanını 1.l ve 2.l anahtarını, veri tiplerini, tabloları ilişkilendirerek sql server üzerinde

kodlarla oluşturunuz.

create database film;

use film;

create table musteri

(

MNo int primary key,

Adi varchar(50),

Sadi varchar(50),

Cinsiyet bit,

CocukSay int,

Egitim bit

);

create table FilmTip

(

FTipiID int primary key,

FTipAdi varchar(50)

);

create table Film

(

FID int primary key,

Fadi varchar(50),

FTipID int foreign key references FilmTip(FTipiID),

Odunc bit,

);

create table OyuncuYon

(

OYNo int primary key,

OyAdSoy varchar(50),

OyYon varchar(20)

)

create table MusteriOyuncu

(

MNo int foreign key references musteri(MNo),

OyNo int foreign key references OyuncuYon(OYNo)

)

create table musteriFilm

(

MNo int foreign key references musteri(MNo),

FID int foreign key references Film(FID),

AlTrh datetime,

GTrh datetime,

Geldi bit

)

create table OyuncuFilm

(

FID int foreign key references Film(FID),

OyNo int foreign key references OyuncuYon(OYNo)

)



2. Çocuk sahibi olan müşterilerden hiç çocuk filmi almayanların isim ve soyisimlerini listeleyen sorguyu yazınız.

select musteri.Adi,musteri.Sadi from musteri

inner join musteriFilm on musteri.MNo=musteriFilm.MNo

inner join film on film.FID=musteriFilm.FID

inner join FilmTip on FilmTip.FTipiID=Film.FTipID

where CocukSay>0 and Film.FTipID=7

group by musteri.Adi,Musteri.Sadi;

3. Şimdiye kadar 5'den fazla bilim kurgu filmi alan müşterilerin isim ve soyisimleri ile birlikte kaç tane bilim kurgu filmi

aldıklarını listeleyen sorguyu yazınız.

select musteri.Adi,musteri.Sadi,count(musteriFilm.FID)

as 'Adedi' from musteri

inner join musteriFilm on musteri.MNo=musteriFilm.MNo

inner join film on film.FID=musteriFilm.FID

inner join FilmTip on FilmTip.FTipiID=Film.FTipID

where film.FTipID=5

group by musteri.Adi,Musteri.Sadi

having count(musteriFilm.FID)>4;

4. Kevin Costner isimli yönetmeni seven ve bu yönetmene ait filmleri alan müşterilere ait bilgileri listeleyen sorguyu

yazınız.

select musteri.Adi,musteri.Sadi from musteri

inner join musteriFilm on musteri.MNo=musteriFilm.MNo

inner join film on film.FID=musteriFilm.FID

inner join FilmTip on FilmTip.FTipiID=Film.FTipID

inner join OyuncuFilm on film.FID=OyuncuFilm.FID

inner join MusteriOyuncu on MusteriOyuncu.MNo=musteri.MNo

where OyuncuFilm.OyNo=5 and MusteriOyuncu.OyNo=5

group by musteri.Adi,Musteri.Sadi;

5. Her bir müşterinin hangi tip filmlerde kaçar adet ödünç aldığını listeleyen sorguyu yazınız.

select musteri.Adi,musteri.Sadi,FilmTip.FTipAdi,

count(musteriFilm.FID) as 'Adet' from musteri

inner join musteriFilm on musteri.MNo=musteriFilm.MNo

inner join film on film.FID=musteriFilm.FID

inner join FilmTip on FilmTip.FTipiID=Film.FTipID

group by musteri.Adi,Musteri.Sadi,FilmTip.FTipAdi;

6. Bugüne kadar aldığı filmleri getirmeyen müşterilerin hangi filmin, hangi yönetmenin filmi olduğunu listeleyen sorguyu bir view kullanarak oluşturunuz.

select film.Fadi,OyuncuYon.OyAdSoy from film

inner join musteriFilm on musteriFilm.FID=film.FID

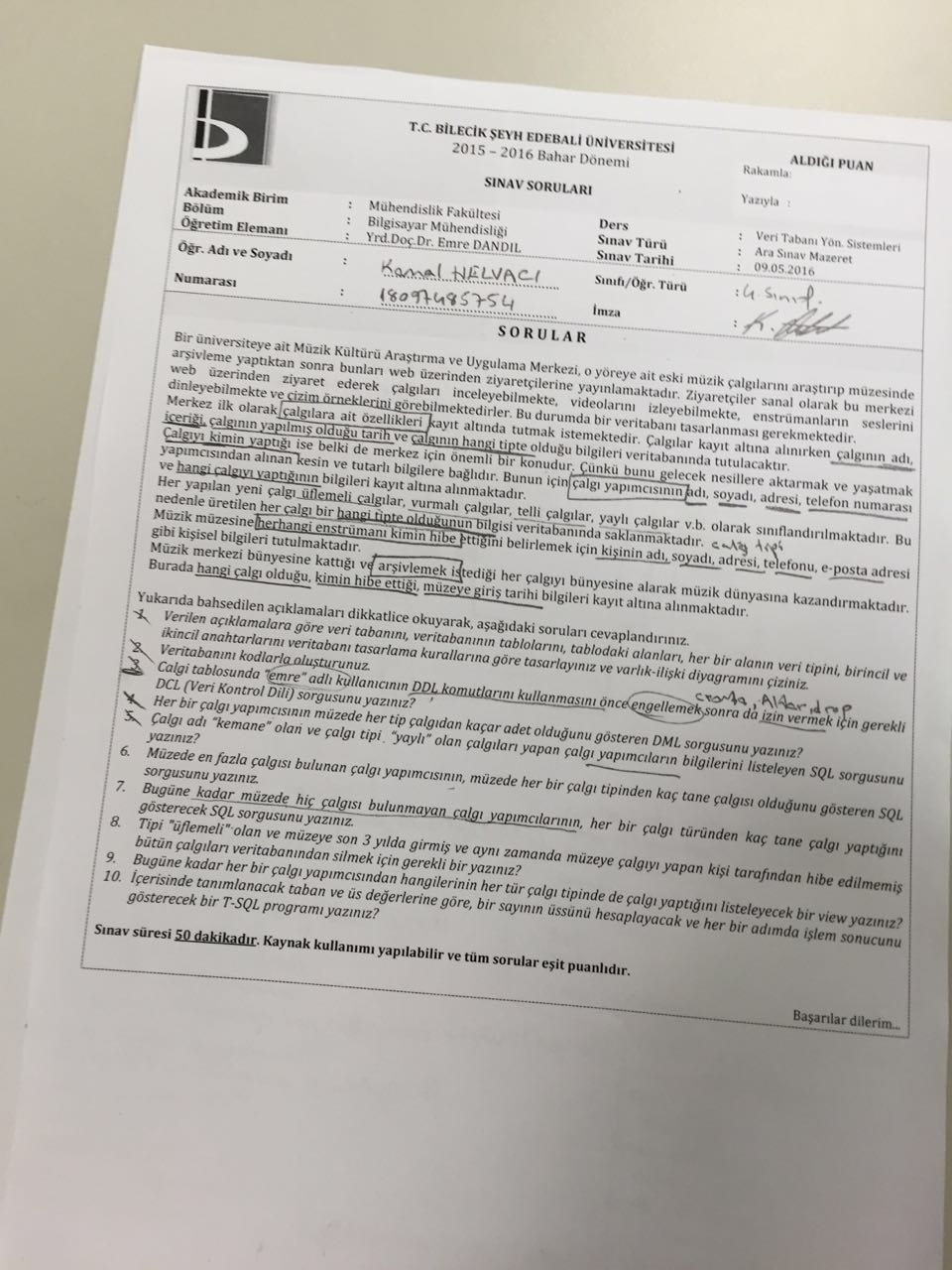
inner join OyuncuFilm on film.FID=OyuncuFilm.FID

inner join FilmTip on FilmTip.FTipiID=Film.FTipID

inner join OyuncuYon on OyuncuYon.OYNo=OyuncuFilm.OyNo

where musteriFilm.Geldi='False' and OyuncuYon.OyYon='1'

group by film.Fadi,OyuncuYon.OyAdSoy;



create database muzikk

use muzikk

create table Calgi(

CNo int primary key identity(1,1),

CAdi varchar(20),

Cİcerik varchar(20),

Ctarih date,

CTNo int foreign key references CalgiTip(CTNo)

)

drop table CalgiTip

insert into Calgi values ('Klarnet','Hüsnü','01/10/2014',2)

insert into Calgi values ('Ney','Hüsnü','01/10/2014',2)

insert into Calgi values ('Gitar','Duman','01/11/2014',3)

insert into Calgi values ('Saz','Duman','01/11/2014',3)

insert into Calgi values ('Davul','Mustafa','01/12/2014',1)

insert into Calgi values ('Darbuka','Mustafa','01/12/2014',1)

create table CalgiTip(

CTNo int primary key identity(1,1),

CTTur varchar(20),

)

insert into CalgiTip values ('Vurmalı')

insert into CalgiTip values ('Üflemeli')

insert into CalgiTip values ('Basmalı')

create table CalgiYapim(

CYNo int primary key identity(1,1),

CYAdi varchar(20),

CYSoyadi varchar(20),

CYAdres varchar(20),

CYTel varchar(20),

CYEposta varchar(20),

Ctarih date,

CNo int foreign key references Calgi(CNo),

CTNo int foreign key references CalgiTip(CTNo)

)

insert into CalgiYapim values ('Yılmaz','Erdogan','Avcılar/İstanbul','02221455786','xxx@xxx','01/10/2014',2,1)

insert into CalgiYapim values ('Berkecan','Erdogan','Avcılar/İstanbul','02221455786','xxx@xxx','01/10/2014',2,1)

insert into CalgiYapim values ('Onur','Onder','Yıldırım/Bursa','02244559786','yyy@yyy','08/10/2015',3,2)

insert into CalgiYapim values ('Aslıcan','Onder','Yıldırım/Bursa','02244559786','yyy@yyy','08/10/2015',3,3)

insert into CalgiYapim values ('Aslı','Enver','Şişli/İstanbul','02125455786','zzz@zzz','01/10/2016',1,1)

insert into CalgiYapim values ('Ayşenur','Enver','Şişli/İstanbul','02125455786','zzz@zzz','01/10/2016',1,2)

Create Table MuzikHibe(

MHNo int primary key identity(1,1),

MHAdi varchar(20),

MHSoyadi varchar(20),

MhAdres varchar(20),

MhTel varchar(20),

CNo int foreign key references Calgi(CNo)

)

insert into MuzikHibe values ('Cem','Adrian','Etiler/İstanbul','02166478569',1)

insert into MuzikHibe values ('Zek','Adrian','Etiler/İstanbul','02166478569',1)

insert into MuzikHibe values ('Sezen','Aksu','Bağcılar/İstanbul','02164578475',2)

insert into MuzikHibe values ('Ali','Aksu','Bağcılar/İstanbul','02164578475',2)

insert into MuzikHibe values ('Murat','Boz','Bostancı/İstanbul','02164343343',3)

insert into MuzikHibe values ('Burhan','Boz','Bostancı/İstanbul','02164343343',3)

Create Table Arsiv(

ANo int primary key identity(1,1),

CNo int foreign key references Calgi(CNo),

MHNo int foreign key references MuzikHibe(MHNo),

CYNo int foreign key references CalgiYapim(CYNo),

CTNo int foreign key references CalgiTip(CTNo),

GirisTarih date,

)

insert into Arsiv values (1,3,1,1,'01/05/2016')

insert into Arsiv values (1,2,1,2,'01/05/2016')

insert into Arsiv values (1,3,2,3,'01/05/2016')

insert into Arsiv values (2,3,2,3,'01/05/2016')

insert into Arsiv values (2,1,3,1,'01/05/2016')

--soru3 tabloda emre kullanıcısını engellemek sonra izn vermek için gerekli sql kodu

--kullanıcının veritabanı veya nesneler üzerinde işlem yapmasını engeller.

Deny All On Yarisma to user1

DENY INSERT,UPDATE,DELETE,SELECT,Create Table on Yarisma to user1

--3)REVOKE ifadesi

--engellenen tüm etkileri kaldırır.

REVOKE ALL from public

REVOKE select on Yarisma from public --seçim için engellenen yetkileri kaldırır.

--soru4: her bir çalgı yapımcısının müzede her tip çalgıdan kaçar tane olduğunu gösteriniz.

select distinct CalgiYapim.CYNo,CYAdi,count(1) from CalgiTip,CalgiYapim,Arsiv where CalgiYapim.CYNo=Arsiv.CYNo

and CalgiTip.CTNo=Arsiv.CTNo group by CalgiYapim.CYNo,CYAdi

select \* from Arsiv

select \*from MuzikHibe

--soru5 çalgı adı klarnet ve çalgı türü üflemeli olan çalgıları yapan çalgı yapımcılarının bilgilerini listele

Select CalgiYapim.CNo,CalgiYapim.CYAdi from CalgiYapim,CalgiTip,Calgi where Calgi.CAdi='Klarnet' and CalgiTip.CTTur='Üflemeli' and calgi.CTNo=CalgiTip.CTNo and

CalgiYapim.CTNo=CalgiTip.CTNo and Calgi.CNo=CalgiYapim.CNo

--soru6 müzede en fazla çalgısı bulunan çalgı yapımcısının müzede her bir çalgı tipinden kaçar tane çagısı olduğunu gösteren sql kodu

select CalgiYapim.CYNo, CalgiYapim.CYAdi from Arsiv,CalgiYapim,CalgiTip Where CalgiYapim.CYNo=Arsiv.CYNo and CalgiTip.CTNo=Arsiv.CTNo and Calgitip.CTNo= (Select max(Arsiv.CTNo) from Arsiv) group by CalgiYapim.CYNo, CalgiYapim.CYAdi

--soru7 bugüne kadar müzede hiç çalgısı bulunmayan çalgı yapımcılarının her bir çalgı türünden kaç tane çalgı yaptığını gösteren sql kodu?

select CalgiYapim.CYNo,CalgiYapim.CYAdi from CalgiYapim,Arsiv,CalgiTip where CalgiYapim.CYNo=(SELECT \* FROM CalgiYapim WHERE CalgiYapim.CYNo!=Arsiv.CYNo)

group by CalgiYapim.CYNo, CalgiYapim.CYAdi

and CalgiTip.CTNo=CalgiYapim.CTNo and Arsiv.CTNo!=CalgiTip.CTNo

**Alt Sorgular**

Bir select sorgusundaki kısıtlama, başka bir tablodaki verilere göre yapılmak istenirse aşağıdaki gibi bir sorgu yapısı kullanılabilir.

SELECT \* [FROM](http://www.btdersleri.com/ders.aspx?konu=156)tablo1

WHERE alanX=

( SELECT alanY FROM tablo2

WHERE alanY=deger )

Burada önce içteki sorgu çalışıp tablo2'den bir değer döndürecek, ardından dıştaki sorgu çalışacaktır.

Örnek: sinavlar tablosunda birinci sınav notu, sınıf ortalamasından daha yüksek olan kayıtları listelemek için şu sorgu kullanılabilir.

SELECT adi, soyadi [FROM](http://www.btdersleri.com/ders.aspx?konu=156) sinavlar

WHERE sinav1 >= ( SELECT AVG(sinav1) FROM sinavlar )

Örnek: Aşağıdaki sorgu çalıştırıldığında;

İçteki sorgu tarafından maasi 2000 ve üzeri olan çalışanların numaraları döndürülür, ardından dıştaki sorgu tarafından personelNo alanı bu değerlerden biri olan kayıtlar listelenir.

SELECT adi, soyadi [FROM](http://www.btdersleri.com/ders.aspx?konu=156) personel

WHERE personelNo IN

( SELECT personelNo FROM ucretler

WHERE maasi >= 2000)

İçteki döngünün çok sayıda değer döndüreceği zamanlarda In, Any ve All komutları sıkça kullanılmaktadır.

**In Komutu**

In komutu belirtilen değerlerden herhangi birine eşit olan kayıtları listelemeyi [sağlar](http://www.btdersleri.com/ders.aspx?konu=156).

Örnekte doğum yeri istanbul, Ankara ya da İzmir olan öğrencilerin listelenmesi sağlanmıştır.

SELECT \* FROM ogrenciler WHERE dgYeri IN [(](http://www.btdersleri.com/ders.aspx?konu=156)"İstanbul", "Ankara", "İzmir")

**Any Komutu**

**>Any** komutu verilen değerlerin herhangi birinden büyük olan (en küçüğünden büyük) kayıtları listelemeyi sağlar.

**<Any** komutu ise verilen değerlerin herhangi birinden küçük olan (en büyüğünden küçük) kayıtları listelemeyi sağlar.

**=Any** komutu ise verilen değerlerden herhangi birine eşit olan kayıtları listelemeyi sağlar. [**>=Any**](http://www.btdersleri.com/ders.aspx?konu=156) ve **<=Any** gibi de kullanılabilir.

Örnek: En düşük maaşlı Mühendisin maaşından daha fazla maaş alan personellerin ad ve soyadlarını listelemek için:

SELECT adi, soyadi FROM ucretler

WHERE maasi >ANY

( SELECT maasi [FROM](http://www.btdersleri.com/ders.aspx?konu=156) ucretler

WHERE gorevi="Mühendis" )

**All Komutu**

Belirtilen şartların tamamının sağlanması istendiğinde kullanılır.

**>All** komutu verilen değerlerin hepsinden büyük olan (en büyüğünden büyük) kayıtları listelemeyi sağlar.

**<All** komutu ise verilen değerlerin hepsinden küçük olan (en küçüğünden küçük) kayıtları listelemeyi sağlar.

**>=All** ve [**<=All**](http://www.btdersleri.com/ders.aspx?konu=156) gibi kullanımları da vardır.

Örnek: En yüksek maaşlı Mühendisin maaşından daha fazla maaş alan personellerin ad ve soyadlarını listelemek için:

SELECT adi, soyadi FROM ucretler

WHERE maasi >ALL

[(](http://www.btdersleri.com/ders.aspx?konu=156) SELECT maasi FROM ucretler

WHERE gorevi="Mühendis" )

**Count Komutu:**

Tabloda bulunan kayıt sayısını öğrenmek isterseniz, COUNT komutunu kullanmalısınız.

**SELECT Count (\*)**  
**FROM Ogrenciler**

Bu SQL kodları ile tablonuzda kaç kayıt olduğunu bulabilirsiniz.

**Group By Komutu:**

Group By komutu çok önemli bir komuttur. Örneğin farklı yerlerden yapılan ödemelerini tek bir ekrandan görmek istersek bu komutu kullanabiliriz.

**SELECT Sehir, Count (\*), Sum(BorcUcreti) from Ogrenciler Group By Sehir**

Örnek olarak yukarıdaki kod kümesini yazarsanız sonucu görebilirsiniz.

**Having Komutu**

Örneğin Ankara ilinden 2, diğer illerden 1 kişi olan bizim örnek tablomuzda bulunan verilere göre işlem yapalım. Having komutu şu işe yarar. Sorgu sonucu dönen değerlere göre (şehir ismi) örneğin 5 ve 5 den fazla o şehirden işlem varsa bunları göstermek için kullanılır.

Örnek:

**SELECT Sehir, Count (\*), Sum(BorcUcreti) from Ogrenciler Group By Sehir Having Count(\*)>=2**

İşte bu komut sayesinde, şehir verisi 2 kez var olan verileri listeler. Bizim tablomuzda 2 veri girişi yapılan sadece Ankara olduğu için bizim örneğimizde sonuç olarak sadece Ankara’yı gösterecek. Tablomuza farklı iller eklesek ve farklı verilerle işlem yapsaydık, 2 ve 2’den fazla adı geçen illeri listeleyecekti.

**Where Komutu**

Where komutu en önemli SQL komutlarındandır. Verileri belirli bir amaca göre sıralar. Örneğin borcu 500 TL’den büyük olanlar… Ya da memleketi Kırşehir olanlar gibi…

**SELECT \* From Ogrenciler**  
**Where Sehir=’Ankara’**

Bu gördüğünüz SQL kodları ile birlikte, Şehir verisinde Ankara olanları listeler…

Peki borcu 1000 TL’den büyük olanları listelemek isteseydik ne yapacaktık?

**SELECT \* From Ogrenciler**  
**Where BorcUcreti>=1000**

Bu SQL kodları ile de borcu 1000 TL’den büyük olanları listeleyebilirdim… Listeleme işleminde sayısal ifadeler ile sözel ifadeye göre işlem yaparken farklılık vardır. Sözel ifadeler tırnak işareti arasında belirtilir…

**And Komutu**

Where komutu ile iki şartlı işlem yapmak istersek, and komutunu kullanmamız gerekir…

Örneğin: Şehir Ankara olsun ve borcu 1000’den büyük olanlar listelensin…

**SELECT \* From Ogrenciler**  
**Where BorcUcreti>=1000**  
**And**  
**Sehir=’Ankara’**

Bu kodları SQL ile yazarsanız, söylediğimiz gibi bu iki şartı da sağlayan kişi ya da kişileri listeleyecektir.

**View Nedir?**

**SQL de,  View temel olarak sanal tablo(geçici ile sanal tablo farklı kavramlar lütfen karıştırmayın) diye geçiyor. Satır ve sütunları olan gerçek tablo gibidir. İçerisinde sadece SQL Fonksiyonlarını, Join, Group By ve Where ifadelerini kullanabilirsiniz. Yapamadıkları yaptıklarından daha az bu yapamadıklarını yazımın en son bölümünde yer vereceğim. (örnekleri hayali tablolar üzerinde anlatacağım). SQL Query açıp aşağıdaki örnek kodları yazıp daha sonra execute,kısayolu f5, ettikten sonra çalışmış olacaktır.**

**View Oluşturma (Create View)**

**Oluşturulan View ler veritabanınız içerisinde “Views” seçeneceğinin altında yer almaktadır. Oluşturalım:**

***CREATE VIEW OnSekizYasindanBuyukKullanicilar***

***AS***

***SELECT kullaniciAdi, YasadigiYer,Yas FROM Kullanicilar WHERE Yas>17***

***Kullanımı:***

***Aynı tabloyu nasıl kullanıyorsan View ide öyle kullanacağız.***

***Select \* From OnSekizYasindanBuyukKullanicilar Where kullaniciAdi like ‘A%’***

**Şifreli View Oluşturma (With Encryption)**

**CREATE VIEW SifreliView**

**With Encryption**

**AS**

**SELECT kullaniciAdi, YasadigiYer,Yas FROM Kullanicilar WHERE Yas<17**

**Not: Şifrelediğiniz View’in kodlarını kendimizde göremeyiz yani Design sayfası bulunmamaktadır. Bunu unutmayı o yüzden yedek almakta fayda var kodlarınızı. Ayrıca proc, function, triggerlarıda şifreleyebilirsiniz.**

**View Güncelleştirme (Alter View)**

**Güncelleştirmek için *alter*ifadesinden sonra güncelleştirmek istediğimiz türü seçiyoruz burada *view*oluyor ve en son olarak*View adını* yazıyoruz. Kodları değiştirten sonra çalıştırmayı unutmayın.**

***ALTER VIEW OnSekizYasindanBuyukKullanicilar***

***AS***

***SELECT kullaniciAdi, Yas FROM Kullanicilar WHERE Yas>17***

**View Silme (Drop View)**

**Aslında Create, Alter, Drop SQL ifadeleri olduğu için tablo, view, fonsiyon, procedur işlemlerinde aynı şekide kullanılıyor, yani view e özel durum söz konusu değil.**

***DROP VIEW OnSekizYasindanBuyukKullanicilar***

**View larda Ne Yapılmaz**

**Yukarıda söylediğim gibi aslında view lerin yapamadıkları daha fazla.**

* **Order  By kullanamazsınız.**
* **İsimsiz kolon bırakamazsınız. Örneğin Sum fonsiyonunu kullanacaksınız As ile kesin bir isim vermek zorundasınız.**
* **Birden fazla sorgu yazamazsınız yani sadece bir select ile başlayan bir cümle yapabilirsiniz.**
* **T-SQL kullanamazsınız**
* **Insert, Update veya Delete kullanamazsınız**
* **Herhangi bir parametre yollayamazsınız.**

Create Database Yarisma2

use Yarisma2

Create Table İl

(

İlPNo int Primary Key,

İlAdi varchar(30),

)

Create Table Uye

(

UNo int Primary Key identity (1,1),

UAdi varchar(30)NOT NULL,

UASoyadi varchar(30)NOT NULL,

UBTrh datetime Null,

İlPNo int foreign key references İl(İlPNo),

USifre varchar(30)NOT NULL DEFAULT('123abc'),

GirSay int Default(0),

Sonpuan int

)

Create Table SoruTur

(

STNo int Primary Key identity (1,1),

StAdi varchar(30)NOT NULL,

)

Create Table Soru

(

SNo int Primary Key identity (1,1),

SBaslik varchar(30)NOT NULL,

C1 varchar(2),

C2 varchar(2),

C3 varchar(2),

C4 varchar(2),

C5 varchar(2),

STNo int foreign key references SoruTur(STNo),

SPuan int NOT NULL,

UNo int foreign key references Uye(UNo),

)

Create Table İstatistik

(

UNo int foreign key references Uye(UNo),

AldTrh datetime,

ToplamPuan int,

)

insert into İl values(1,'Adana')

insert into İl values(2,'Adıyaman')

insert into İl values(3,'Afyon')

insert into İl values(4,'Ağrı')

insert into İl values(5,'Amasya')

insert into İl values(6,'Ankara')

insert into İl values(7,'Antalya')

insert into İl values(8,'Artvin')

insert into İl values(9,'Aydın')

insert into Uye values('Turgut','Özseven','','1','123',3,50)

insert into Uye values('Sercan','Kaymaz','', '2','123a',0,50)

insert into Uye values('Necati','Tuncer','','3','123ab',2,50)

insert into Uye values('Turgut','Demir','','4','123abc',1,50)

insert into Uye values('Sinan','Karakayalı','','5','123abcd',4,50)

insert into Uye values('Burhan','Özseven','','6','123abcde',3,50)

insert into SoruTur values('Tarih')

insert into SoruTur values('Matematik')

insert into SoruTur values('coğ')

insert into SoruTur values('Fizik')

insert into SoruTur values('Geo')

insert into SoruTur values('Kimya')

insert into Soru values('aaa','a','b','c','d','a',1,10,1)

insert into Soru values('aaa','','','','','a',1,10,2)

insert into Soru values('aaa','','','','','a',3,15,2)

insert into Soru values('aaa','','','','','a',4,10,3)

insert into Soru values('aaa','','','','','a',5,10,4)

insert into Soru values('aaa','','','','','a',5,10,3)

-------------------------------------------------------------------------------

--ALTER İFADESİ

--ekleme

--tabloya yeni sutun ekler.

Alter Table Uye ADD UAdres varchar(30)

--güncelleme sütunun türünü değiştirir.

Alter Table Uye Alter Column UAdres int

--silme

--tablodaki sütunu siler

Alter Table Uye Drop Column UAdres

-------------------------------------------------------------------------------

--DROP İfadesi

--nesne veya tablo veya veri tabanını siler.

--Drop Database Okul

--Drop Table Öğrenci

--\*\*Truncate Table ogrenci(tablonun tüm içeriğini siler.)

--\*\*Delete Table ogrenci(istenilen kayıtları siler)

--delete from Uye where UNo=10 gibi

------------------------------------------------------------------------------

----DML (Veri İşleme Dili)

--1)SELECT İfadesi

select \* from Uye --tabloyu listeler.

select UAdi,UASoyadi from Uye --tablodaki belirli istenilen stenilen sütunları gösterir.

--a)WHERE ifadesi (koşul ifadesi)

select \* from Uye where İlPNo<9

--b)AND,OR ifadesi

select \* from Uye where İlPNo<9 and USifre>=15 --iki koşul sağlanırsa

select \* from Uye where İlPNo<9 or USifre>=15 -- herhangi bir koşul sağlanırsa

--c) IN, NOT IN ifadesi

--fazla şartın bulunduğu sorgularda or yapısından kurtulmak için kullanılır.

select \* from İl where İlPNo IN(1,2,3,4) --plakası 1 veya 2 veya 3 veya 4 olanı gösterir

select \* from İl where İlPNo NOT IN(1,2,3,4)

--d) LIKE ifadesi

--Kayıt arama esnasında kullanılır.

select \* from Uye where UAdi LIKE('\_\_\_\_M')--beşinci harfi M olan uyeleri listeler

select \* from Uye where UAdi LIKE('\_\_\_\_a\_')--beşinci harfi A olan uyeleri listeler.

select \* from Uye where UAdi LIKE('%M') --son harfi m olan

select \* from Uye where UAdi LIKE('M%')

select \* from Uye where UAdi LIKE('%M%')

select \* from Uye where UAdi LIKE('B\_\_\_\_')

-------------------------------------------------------------------------------------------------------

--2) INSERT İfadesi

--tablolara yeni veri girişi için kullanılır.

insert into İl values(80,'Osmaniye')

--------------------------------------------------------------------------------------------

--3)UPDATE ifadesi

--tablo içindeki var olan kayıtları değiştirmek,güncellemek için kullanılır.

update il set İlAdi='Düzce' where İlPNo=81 --plakası 81 olan ilin adını düzce olarak güncellememizi sağlar

select \* from İl

--4)DELETE ifadesi

--tablonun içerisindeki kayıtların tamamını veya belirli koşula uyanları siler.

delete from Soru where SNo=10

------------------------------------------------------------------------------------------------------------

--DCL (Veri Kontrol Dili)

--1) GRANT ifadesi

--Kullanıcıya çeşitli iznler vermek için kullanılır.

grant All On Yarisma to user1 --kullanıcıya tüm izimleri verir.

grant create table on Yarisma to user1 --kullanıcıya sadec tablo oluşturma izni verir.

--2)DENY ifadesi

--kullanıcının veritabanı veya nesneler üzerinde işlem yapmasını engeller.

Deny All On Yarisma to user1

DENY INSERT,UPDATE,DELETE,SELECT,Create Table on Yarisma to user1

--3)REVOKE ifadesi

--engellenen tüm etkileri kaldırır.

REVOKE ALL from public

REVOKE select on Yarisma from public --seçim için engellenen yetkileri kaldırır.

-----------------------------------------------------------------------------------------------------------------

--TEK TABLO İÇİNDE SORGULAMALAR

Select Uye.UAdi,Soru.SBaslik from Uye,Soru where Uye.UNo=Soru.UNo

--1) DISTINCT >> tekrarlanan satır ve sütünları engeller

Select distinct UAdi From Uye

-----------SORGU SONUCUNU SIRALAMA---------

--2 ORDER BY>>>> tabloda kayıtların sıralanmasını sağlar.

Select UNo From Uye order by UNo asc -->>>>> desc büyükten küçüğe sıralar

Select UNo From Uye order by UNo desc -->>>>> asc küçükten büyüğe sıralar

Select \*From Uye where UAdi LIKE '%m' order by UAdi -->>>>> desc büyükten küçüğe sıralar

--BETWEEN...and yapısı >>>> birden fazla şartı sıralar

Select \*from Uye where UAdi LIKE 'B%' or UNo between 5 and 7

--AS ifadesi >>>>>> Sütüna takma isim vermek için veya ismini değiştirmek için

Select UAdi AS "üyeadı" from Uye

----- Sütünlar üzerinde aritmetiksel işlemler--------

Select SPuan, Spuan+10 as "name" from Soru

-----Süttünların birleştirilmesi--------------

Select UNo, (UAdi +' '+ UASoyadi) as "adsoyad" from Uye

--\*\*\*>>> KOŞULA BAĞLI SORGULAMARDA WHERE VE ORDER BY KULLANILACAKSA ÖNCE WHERE YAZILMALI

Select \*from Uye where UNo>=3 order by UNo -- önce where

Select 9\*8 as "name"

------------FONKSİYONLAR-----------

--1 SUBSTRİNG() >>> Veri kümsei içindeki bilgilerin başlangıç olarak belirtilen

---pozisyondan başlayarak belirtilen karakter sayısı kadar alınmasını sağlar.

Select UNo, SUBSTRING(UAdi,2,5) as "yeniadd", UAdi as "yeniad" from Uye order by yeniadd desc

--2 LOWER(), UPPER() ifadeleri >> lower verilern değeri küçük harfe, upper ise büyük harfe çevirir.

Select UPPER(İlAdi) from İl

Select lower(İlAdi) from İl

-----TRIM() , RTRIM , LTRIM, >>> Boşlukları kaldırır.

Select RTRIM(UAdi) as UAdi,UASoyadi from Uye --sağındaki boşluğu kaldırır

--4 LEN() , LENGTH() >>>> karakter sayısını bulma

select UAdi,len(UAdi) as "karakter sayisi" from Uye

--5 replace() yer değiştirme

select UAdi,UASoyadi,REPLACE(UAdi,'Ali','aliiiiiiiii') from Uye

--6 REVERSE() >>>> tersten yazdırma

select UAdi,REVERSE(upper(UAdi)) from Uye

select UAdi,REVERSE(UAdi) as "yeni",upper(UAdi) from Uye

--7 ABS() >>> mutlak değer alma

Select ABS(-1.25) as "amk"

--8 FLOOR(), CEILING() >>> aşağı-yukarı yuvarlama

Select FLOOR(1.256)-- AŞAĞI YUVARLAR

Select CEILING(1.256)-- YUKARI YUVARLAR

--9 ROUND() >>>> en yakın değere yuvarlar

Select round(1.26,1)

-- POWER() >>> üst alma

SElect power(2,3)

--10 SQRT() karekök

Select sqrt(4)

Select round(sqrt(40),0)

--11 SUM() toplam bulma

select sum(Spuan) from soru

--12 AVG() ortalama değer hesaplama

Select round(AVG(Spuan),2) from Soru

---13 Max min

Select MAX(Spuan) from Soru

Select MIN(Spuan) from Soru

---14 count() >>>> satır sayısını bulur

Select count(Spuan) from soru

--15 -- GETDATE güncel saat bilgisini verir

select GETDATE()

select DATEPART(YEAR,GETDATE())

select DATEPART(DAY,GETDATE())

select DATEPART(HOUR,GETDATE())

select DATEPART(MINUTE,GETDATE())

--------------------------BÖLÜM 8: GRUPLANDIRARAK SORGULAMA---------------------------------

--1) >>>GROUP BY

--\*\*\* WHERE koşulundan sonra ve ORDER BY ifadesinden önce kullanılmalı.

--GRUPLANDIRMA İŞLEMİNDE KULLANILAN FONKSİYONLAR

--a) AVG() ortalamaya göre gruplandırma

select UNo,AVG(SPuan) as ort from Soru where UNo>=3 Group By UNo order by UNo desc

--b) Count() grupların içerdikleri satır sayısımı verir

select UNo,Count(SPuan) as satirsayisi from Soru Group By UNo

select UAdi,Count(UAdi) as satirsayisi from Uye Group By UAdi

--C) Max() gruptaki sütünün maxını bulur.

select UNo ,MAX(SPuan) as maxı from Soru Group By UNo

--d) MIN() mini bulur

select UNo ,MIN(SPuan) as mini from Soru Group By UNo

--e) SUM() toplamını verir.

select UNo,SUM(SPuan) as toplam from Soru Group By UNo

--BİRDEN FAZLA SÜTUNA GÖRE GRUPLANDIRMA

select SBaslik,UNo ,MAX(SPuan) as maxı,MIN(SPuan) as mini from Soru Group By UNo,SBaslik

-- GRUPLANDIRMADA KOŞUL İFADELERİ(HAVİNG)

--\*\*\*HAVING Group By dan sonra order by dan önce kullanılır.

select SBaslik ,(MAX(SPuan) -MIN(SPuan)) as Fark from Soru Group By SBaslik Having (MAX(SPuan) -MIN(SPuan))>5

-----------------------------------BÖLÜM:9 BİRDEN FAZLA TABLO ÜZERİNDE SORGULAMA----------------------------

---A) Tablolara Takma İsim Kullanma

select distinct uyem.UAdi,sorum.SBaslik from uye uyem,soru sorum where uyem.UNo=sorum.UNo

select distinct Uye.UNo,Soru.SNo,Soru.C1 from Uye ,Soru where Uye.Uno=Soru.UNo and soru.C1='a' and Uye.UNo in (select Uno from Soru)

--B)Tabloların Birleştirilimesi(JOİN)

--Inner Join(iç Birleştirici)

--Outer Join(Dış Birleştirici)

--Cross Join(Çapraz Birleştirici)

-------------------------------------2.KİTAP BÖLÜM:2 TSQL---------------------------------------------------

declare @sayi1 int,@sayi2 int, @toplam int

set @sayi1=15

set @sayi2=10

set @toplam=@sayi1+@sayi2

print('Toplam:' + CAST(@toplam as varchar(10)))

print(@toplam)

----

declare @eyp int

select @eyp=MAX(Spuan) from Soru

select @eyp as 'en yüksek puan'

select 5/0

select @@ERROR AS 'Hata Kodu'

select @@SERVERNAME AS 'Sunucu İsmi'

select @@SERVICENAME AS'Servis Adı'

select @@VERSION

select @@LANGUAGE

select @@MAX\_CONNECTIONS

--TABLO TİPİ DEĞİŞKENLER

declare @personel Table (

noo int identity,

adsoyad varchar(50) not null,

telefon varchar(15)

)

insert into @personel values('Mehmet Kirazlı','56465465')

insert into @personel values('Eda Elletmez','56465465')

select \* from @personel order by adsoyad

declare @bml TABLE(

bno int,

badi varchar(30)

)

insert into @bml values(1,'Bilgisayar Müh.')

insert into @bml values(2,'Makine Müh.')

insert into @bml values(3,'Kimya Müh.')

select\*from @bml

--

declare @st table

(

STNo int primary key,

STAdi varchar(20) not null

)

use Yarisma

------OUTPUT

--ekleme

insert into SoruTur output inserted.STNo,inserted.STAdi into @st values('manyetik')

select \*from @st

--güncelleme

update SoruTur set StAdi='Manyetik' output inserted.STNo where STNo=11

select \* from @st

--silme

delete from Sorutur output deleted.STNo where STNo=11

select \* from SoruTur

--ilk 3 soruya 10 ar puan daha eklenirken kayıtların ilk ve son durumlarını kullanıcıya gösteren bir output yaznız

declare @soru TABLE(

sno int,sbaslik varchar(30),

stno int,

spuan\_eski int,

spuan\_yeni int,

uno int

)

update Soru set soru.SPuan=Soru.SPuan+10

output

inserted.SNo,inserted.SBaslik,inserted.STNo,deleted.SPuan,inserted.SPuan,inserted.Uno into @soru where soru.SNo in(select Top 3 soru.SNo from Soru)

select \* from @soru

Yarışma sitesi veritabanı üzerinde üye tablosuna yeni bir kayıt eklenirken eklenen bu kaydın kullanıcıya gösterilmesi için

--bir output oluşturunuz.

--insert output

declare @uye TABLE(

uno int,

uadi varchar(20),

usoyadi varchar(30),

plaka varchar(15),

sifre varchar(20)

)

insert into Uye(UAdi,UASoyadi,UBTrh,İlPNo,USifre,GirSay,Sonpuan)

OUTPUT inserted.UNo,inserted.UAdi,inserted.UASoyadi,inserted.İlPNo,inserted.USifre into @uye

values('Orhan','Başak','',1,'123',0,0)

select\*from @uye

GO

--Üye nosu 3ten küçük olan üyeleri veritabanından silerken silinen kayıtları kullanıcıya gösterecek bir output yazınız.

declare @uye2 TABLE(

uno int,

uadi varchar(20),

usoyadi varchar(30),

plaka varchar(15),

sifre varchar(20)

)

delete from Uye

OUTPUT deleted.UNo,deleted.UAdi,deleted.UASoyadi,deleted.İlPNo,deleted.USifre into @uye2 where Uye.UNo>3

select\*from @uye2

--Genel kültür türünde hiç soru göndermemiş üyelerin veritabanından silinmesi esnasında silinen kayıtlara ait üyelerin

--ad ve soyad bilgilerini kullanıcıya gösteren bir output tasarlayanız.

declare @uye3 TABLE(

uno int,

uadi varchar(20),

usoyadi varchar(30),

plaka varchar(15),

sifre varchar(20)

)

declare @uye4 TABLE(

STNo int,

STAdi varchar(50)

)

delete from Uye,SoruTur

OUTPUT deleted.UNo,deleted.UAdi,deleted.UASoyadi,deleted.İlPNo,deleted.USifre into @uye3,@uye4

where SoruTur.StAdi STAdi NOT IN STAdi='Genel'

select\*from @uye3,@uye4

------------------AKIŞ KONROLLERİ---------------

--1)İF ELSE YAPISI

--IF

declare @us int

select @us=Count(Uye.UNo) from Uye

IF(@us>=0 and @us<=5)

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('az sayida uye var.')

end

else if(@us>5 and @us<=20)

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('yeterli sayida uye var.')

end

else

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('fazla uye var.')

end

go

select \*from Uye

Alter Table uye Add cinsiyet bit null

update Uye set Cinsiyet=1

insert into Uye values ('Ezgi','Karahan','',1,'123',2,25,0)

update Uye set cinsiyet=0 where Uno=2

select Uye.UAdi,Uye.UASoyadi,Uye.Cinsiyet,Cinsiyeti=

Case Uye.Cinsiyet

when 1 then 'Erkek'

when 0 then 'Bayan'

end

from uye

select\* from Uye

Alter Table uye Add cinsiyet bit null

update Uye set Cinsiyet=1

insert into Uye values ('Ezgi','Karahan','',1,'123',2,25,0)

update Uye set cinsiyet=0 where Uno=2

select Uye.UAdi,Uye.UASoyadi,Uye.Cinsiyet,Cinsiyeti=

Case Uye.Cinsiyet

when 1 then 'Erkek'

when 0 then 'Bayan'

end

from uye

select\* from Uye

--1 ile 10 arasındaki tüm sayıları yazdıran kod

----@declare @sayac in

declare @sayac int

set @sayac=1

while(@sayac<=10)

begin

print(@sayac)

set @sayac+=1

end

Go

-- bir sayının asal sayı olup olmadığını gösteren bir tsql kodu yaz

declare @sayi int

declare @y int

declare @asal char(10)

set @sayi=11

set @y=1

set @asal='Asal'

while (@y<@sayi-1)

begin

set @y=@y+1

if (@sayi%@y)=0 set @asal='Asal değil'

if (@sayi%@y)=0 break

end

print @asal

------------------

declare @sayii int ,@syc int,@k int

set @sayii=72

set @syc=1

set @k=0

while(@syc<=@sayii)

begin

if(@sayii%@syc=0)

begin

set @k=+1

print('Tam bolunur:'+CAST (@syc as varchar(10)))

end

set @syc+=1

end

if(@k>=3)

begin

print('Bu sayı asal değil:'+CAST (@sayi as varchar(15)))

end

else

begin

print('Bu sayı asal:'+CAST (@sayi as varchar(15)))

end

go

-------------------------Fazlalık-----------------------------------------

Select \*from İstatistik where AldTrh>'03/03/2011'-

Select \*from İstatistik where Month( AldTrh)>5 and year ( AldTrh) >=2011

----------------------------ÇAlışmna----

a Select Model, marka, Fiyat from arac where model<=2004

b Select Model, marka, Fiyat from arac where model<=2009 and Model>=2000

c Select \*from Satış where Sat\_tarih> '01/05/2001' and SFiyat>10000

Select \*from İstatistik where AldTrh>'03/03/2011'

Select \*From Musteri where MAdi Lıke('%y%') and MAdres lıke '%Turhal%'

Select Aracno , model , plaka , fiyar from Arac where (Marka lıke 'Opel%' or lıke 'Fiat%') and Model>2000

Grant Select,Insert,update, delete on MUsteri to A

Deny Select,Insert,update, delete on Araç to A

Grant Select ,insert on musteri to B

Deny Uptade delete On musteri to B

Grant Select,Insert,update, delete on MUsteri to A

grant All On Yarism to user1

select \*from Musteri Order by MAdi

select UASoyadi,UAdi from Soru,Uye where Soru.SNo=Uye.UNo

--Order by UASoyadi desc

Select UAdi,UASoyadi as SAtıs From Uye

select \* from Soru where Soru.UNo in (select Uye.UNo from Uye where Uye.UAdi='ayşe')

8.bölüm çalışmasoruları

select count(UASoyadi) as uyeler from Uye group by UASoyadi having (UASoyadi)='Bilen'

--------------------------------------2.KİTAP BÖLÜM:1 ------ TSQL-----------

-------------TSQL Akış kontrolleri

use Yarisma

select \*from Uye

--IF

declare @us int

select @us=Count(Uye.UNo) from Uye

IF(@us>=0 and @us<=5)

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('az sayida uye var.')

end

else if(@us>5 and @us<=20)

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('yeterli sayida uye var.')

end

else

begin

print('Uye sayisi:'+CAST(@us as varchar(10)))

print('fazla uye var.')

end

go

-----------CONTİNUE

select \*from Uye

declare @sayac int

set @sayac=1

while @sayac<=5

begin

if @sayac=3

begin

set @sayac=@sayac+1

continue

end

print cast(@sayac as varchar(10))+'karesi :'+ cast(@sayac\*@sayac as varchar(10))

set @sayac=@sayac+1

end

--G0 to: İstenilen bi etikete dallanılmasını veya dallanma yapmasını sağlar.

--WAitfor: İstenilen bir anda yada belli bir süre gecikme ile tsql kodlarının çalışmasını sağlar.

Waitfor time '12:00:01' -- bu saatten itibaren çaıştır

Waitfor delay '12:00:01' --12 saat 1 dk sonra çalıştır.

--------------------------------------------------------QUİZ-----------------------------------------------------------------------------

--- yarisma sitesi veri tabanı üzerinde oluşturulacak yeni bir tabloda her bir uyenin bugüne kadar çözmüş olduğu sorular

---ve bu sorulara vermiş olduğu cevaplar tutulacak.yarışma veri tabanı diyagramını göz onunde bulundurarark ve gerekli alanları

---belirleyerek bu tabloyu oluşturun. daha sonra her bir uyenin bugüne kadar kaç tane soru çözdüğünü bu sorulardan kaçına doğru

---cevap verdiğini kaç tanesine yanlış cevap verdiğini,almış olduğu toplam puanı ekranda listeleyecek sql ve tsql kodlarını yazınız.

create table uyebilgi

(

SNo int foreign key references Soru(SNo),

UNo int foreign key references Uye(UNo),

UCevap varchar(5)

)

Alter table Soru Add DC varchar(5)

select UNo,Count(SPuan) as çözdüğüsorusayısı from Soru Group By UNo

declare @ss int,@dcs int,@ycs int,@puan int, @uno int

select @ss= COUNT(uyebilgi.UNo) from uyebilgi

select @dcs= COUNT(uyebilgi.SNo) from Soru,uyebilgi where Soru.DC=uyebilgi.UCevap and uyebilgi.SNo=Soru.SNo

select @puan=@puan+Soru.SPuan from Soru,uyebilgi,Uye where Soru.DC=uyebilgi.UCevap and uyebilgi.SNo=Soru.SNo and Uye.UNo=uyebilgi.UNo

select distinct Uye.UAdi,Uye.UASoyadi,@ss AS 'Çözdüğü Soru Sayısı' , @dcs AS 'Doğru',@ycs AS 'Yanlış' ,@puan from Uye,uyebilgi where Uye.UNo=uyebilgi.UNo and Soru where Uye.Uno=Soru.UNo and uyebilgi.SNo=Soru.SNo

while (@dc=@uc)

begin

set @syc+=1

end

---------------------------------------------------------------------------------------------------------------------------------

-------HATA YAKALAMA----------

Begin Try

........

.... // Hata olabilecek kodlar

........

Begin Catch

....

.... Ne yapılacak

....

end catch

\*/