GameBros Oyuncu ve Oyun Platformu Veritabanı Projesi

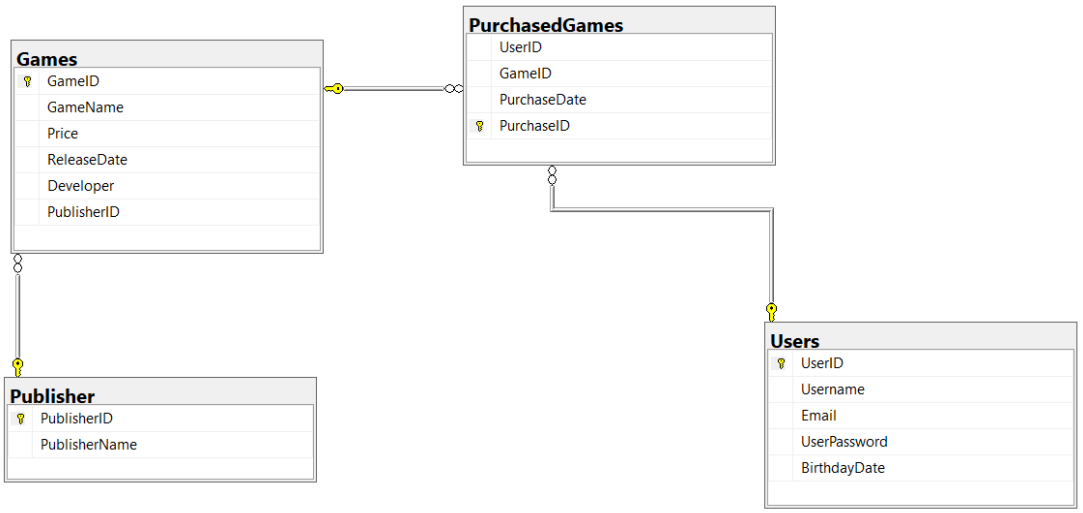
Ömer Faruk Üçer
ofucer99@gmail.com
+90 546 769 8000

1.Proje Amacı ve Özeti

Veritabanı projesi için Microsoft SQL Server veritabanı kullanarak ilişkisel tablolar oluşturmak, tablolara çok sayıda veri eklemek,bu veriler üzerinden işlemler yapmak, stored procedure, trigger, view yapılarından kendi veritabanımıza uygun örnekler eklemek ve kendi belirlediğimiz sorguları yazarak bu sorguları çalıştırmamız istendi.

2.Veritabanı Hakkında ve Veritabanı Özellikleri

Proje için video oyunlarının kullanıcılar tarafından satın alındığı ve oynandığı bir uygulamaya ait bir veritabanı tasarımı gerçekleştirdim. Uygulamada oyunların bulunduğu Games tablosu, Kullanıcıların bulunduğu Users tablosu, satın alınan oyunların bilgisinin tutulduğu PurchasedGames tablosu ve yayıncıların olduğu Publisher tablosu olmak üzere 4 tablodan oluşmaktadır.



Games Tablosu

Bu tabloda oyunlara ait **GameID** (int primary key), **GameName** (nvarchar(50)), **Price** (float), **ReleaseDate** (Date), Developer (nvarchar(50)) ve **PublisherID** (int foreign key) sütunları bulunmaktadır.**GameID** sütunu birincil anahtardır ve her oyunun kendine ait özel bir **GameID'**si vardır. Diğer veriler oyunun adı, fiyatı, yayınlanma tarihi, geliştirici adı gibi sabit bilgileri içermektedir. **PublisherID** değeri int bir değer olup **foreign key**'dir ve bu değer Publisher tablosundaki **PublisherID** değerini referans olarak gösterir. Böylelikle **Games** tablosuyla **Publisher** tablosu arasında ilişkisel bir bağlantı sağlanmış olur. Bu bağlantı sağlandığı için de bu tablolar arasında birleştirme işlemi yapılabilir.

Publisher Tablosu
Bu tablo oyunları yayınlayan yayıncıların isimleri **PublisherName** (nvarchar(50)) ve onlara ait identity **PublisherID** (int primary key) bilgilerinden oluşmaktadır. Buradaki **PublisherID** sütunu **Games** tablosu tarafından referans alındığı için **Games** tablosuyla birleştirilip kullanılabilir.

Users Tablosu
Kullanıcıların kullanıcı adı, email, şifre ve doğum tarih bilgilerini tutan bir tablodur. Sütunlarında **UserID** (int primary key), **Email** (nvarchar(50)), **UserPassword** (nvarchar(50)), **BirthdayDate** (Date) değerleri bulunmaktadır. **UserID** birincil anahtardır ve her kullanıcıya özgüdür. Diğer veriler

PurchasedGames Tablosu
Bu tablo doğrudan bilgi tutmak yerine iki tablo arasındaki ilişkiyi tutmak amacıyla tasarlanmıştır. Platforma üye olan her kullanıcının yaptığı alım işlemleri bu tabloda aldığı oyuna ait **PurchaseID** (int primary key), **UserID** (int foreign key), **GameID** (int foreign key), **PurchaseDate** (Date) bilgileri bulunur.Her oyun alma bilgisinin kendine ait bir **PurchaseID**'si bulunur. **PurchaseID** birincil anahtardır ve her satın alma işlemi için benzersizdir. **UserID** bilgisi oyunu satın alan kullanıcıya ait olup bu sütun **Users** tablosundaki **UserID** sütununu referans olarak alır. Böylece satın alım yapan kişinin kimlik bilgisi doğrudan Kullanıcılar tablosundan getirilirmiş olur. **GameID** sütunu **Games** tablosundaki **GameID** tablosuyla ilişkilidir.Böylece satın alma işleminin hangi oyunla alakalı olduğunu doğrudan **Games** tablosundaki referansını kullanarak elde etmiş oluruz. Bu tablodaki iki foreign key sayesinde bu tablo **Games** ve **Users** tablolarıyla birleştirilebilir.

3 Tablolar Arası İlişkiler
GameBros veritabanındaki tablolar birbirlerine foreign keylerle bağlı oldukları için ilişkisel tablolardır. Yani birbirlerindeki sütunları referans olarak gösterirler. Tablo olarak baktığımızda ise tablolar arasında özel ilişkiler bulunmaktadır.

- **3.1 Games ve Users Tabloları arasındaki ilişki (n-n / çok-çok)**
  Kullanıcılar ve Oyunlar tablolarına baktığımızda şöyle bir yorum yapabiliriz. Her kullanıcı birden fazla oyuna sahip olabilir. Aynı şekilde her oyunun birden fazla kullanıcısı olabilir. Bu ilişki türünden dolayı bu iki tablo arasında n-n (çok-çok) ilişki türü bulunmaktadır.
- **3.2 Games ve Publisher Tabloları arasındaki ilişki (1-n/çok-tek)**
  Bu tablolar arasına baktığımızda ise şöyle bir yorum yapabiliriz. Her Oyunun bir tane yayınlayıcısı vardır (Normalde bir oyunu yayınlayan birden fazla yayıncı olabilir ancak duruma örnek olması açısından sadece bir yayıncı bilgisi eklenmiştir.). Fakat Bir yayıncının birden fazla yayınladığı oyun olabilir. Yani bir tarafta oyunlar tek bir yayıncıya sahip olabilirken yayıncılar birden fazla oyuna sahip olabilirler.Bu ilişki türünden dolayı bu tablolar arasında 1-n (bir-çok) ilişki türü bulunmaktadır.

4.Yazılan Sorgular ve Sonuçları

Sorgu 1: ID'si verilen kullanıcının aldığı oyunların toplam fiyatı

```
------------ID'si verilen bir kullanıcının  aldığı tüm oyunların fiyatları toplamı------------
select Username ,sum(price) as toplam_fiyat
from Users
join PurchasedGames on Users.UserID=PurchasedGames.UserID
join Games on PurchasedGames.GameID=Games.GameID
where Users.UserID=280907
group by Username
```

	Username	toplam_fiyat
1	User30905	30.04

Yazdığım bu sorguda girilen **UserID** bilgisine göre Kullanıcı adı ve aldığı oyunların toplam fiyatlarını yazdırıyor. **Users** tablosundan Username, **Games** tablosundan Price ve bu iki tablonun eşlenmesi için **PurchasedGames** tablosu gerekiyor. Sorgu yazılırken bu üç tablonun birleştirilme işlemi olması gerektiği için sorguda iki **join** ifadesi kullanılırken **Username** özelliğine göre gruplanması için group by ifadesi kullanıldı.Ayrıca Oyun fiyatlarının

```
select Username,GameName,Price
from Users
join PurchasedGames on Users.UserID=PurchasedGames.UserID
join Games on PurchasedGames.GameID=Games.GameID
where Users.UserID=280907
```

toplamını elde etmek için **Price** sütunundaki verileri **sum** fonklsiyonu yardımıyla toplattırdım.Sorgunun doğruluğunu test etmek için Bu kullanıcının aldığı oyunları ve fiyatları yazdıran şöyle bir sorgu daha yazdırırsak
Böyle bir çıktı elde ederiz.Elde ettiğimiz çıktıdaki fiyatları topladığımızda gerçekten 30.04$ dolar elde ettiğini görebiliriz.

	Username	GameName	Price
1	User30905	PataNoir	6.99
2	User30905	Enigmatis: The Ghosts of Maple Creek	6.99
3	User30905	Margot's Word Brain	2.09
4	User30905	Heaven Island Life	0.79
5	User30905	Passage 4	5.99
6	User30905	Haunted Halls: Revenge of Doctor Blackmore Colle...	7.19

Sorgu 2: Her Bir Yayıncının Yayınladığı Toplam Oyun Sayısı ve Oyunların Toplam Fiyatları

```
------------Her Bir Yayıncının Yayınladığı Toplam Oyun Sayısı ve Oyunların Toplam Fiyatları---
select PublisherName,count(*) as yayinladigi_toplam_oyun_sayisi,sum(Price) as toplam_fiyat
from Games
join Publisher on Games.PublisherID=Publisher.PublisherID
group by PublisherName
order by yayinladigi_toplam_oyun_sayisi desc
```

	PublisherName	yayinladigi_toplam_oyun_sayisi	toplam_fiyat
1	Big Fish Games	211	1440.49000000001
2	Strategy First	136	1038.75
3	Ubisoft	107	1690.87
4	Square Enix	99	1204.28
5	THQ Nordic	98	1259.33
6	Sekai Project	96	943.110000000001
7	Choice of Games	94	362.06
8	Dagestan Technology	91	249.49
9	1C Entertainment	89	555.93
10	SEGA	77	1028.3
11	Degica	77	960.530000000001
12	Plug In Digital	74	484.4
13	Slitherine Ltd.	73	1605.28
14	KISS ltd	71	410.2
15	AGM PLAYISM	69	406.87
16	Artifex Mundi	67	529.93
17	Buka Entertainment	66	316.34

Bu sorguya baktığımızda sorguda bizden istenenler tüm yayıncıların ismi, yayınladığı toplam oyun sayısı ve yayınladıkları oyunların toplam fiyatları. Bu bilgilerden yola çıkarak yayıncı isimleri için **Publisher** tablosu ve fiyat toplamını bulmak için **Games** tablolarının birleştirilmesi gerekiyor. Yapılan birleşimler sonrası toplam oyun sayısını **count** ile seçerken toplam fiyat **sum(Price)** ile seçilerek ekrana yazdırılıyor ve en sonunda yayıncıların yayınladığı oyunların çoktan aza olduğu bir sıralamayı **Order By** ve **desc** deyimleri yardımıyla ekrana yazdırılıyor.

Sorgu 3: Aldığı Oyunun Yayınlanma Tarihi ile Aynı Yıl ve Ayda Doğan Kullanıcılar

```
select Users.UserID,UserName,BirthdayDate,GameName,ReleaseDate
from Users
join PurchasedGames on Users.UserID=PurchasedGames.UserID
join Games on PurchasedGames.GameID=Games.GameID
where DATEDIFF(year,BirthdayDate,ReleaseDate)=0
and DATEDIFF(month,BirthdayDate,ReleaseDate)=0
```

	UserID	UserName	BirthdayDate	GameName	ReleaseDate
1	296167	User46165	2011-02-18	Drakensang: The River of Time	2011-02-16
2	255004	User5002	2008-08-22	Sherlock Holmes - Nemesis	2008-08-07
3	256241	User6239	2009-04-23	X-Blades	2009-04-30
4	297853	User47851	2009-07-17	King's Quest™ Collection	2009-07-23
5	294986	User44984	2009-10-20	Painkiller: Resurrection	2009-10-27
6	280971	User30969	2010-10-02	Aura: Fate of the Ages	2010-10-04
7	292728	User42726	2010-02-18	Secret of the Magic Crystals	2010-02-03

Bu sorguda Alınan oyun bilgisi olduğundan **PurchasedGames** tablosu, Oyun Adı ve Yayınlanma tarihi bilgileri oluğu için **Games** tablosu, Kullanıcı adı ve doğum günü tarihi bilgileri olduğu için de **Users** tablosuna ihtiyaç duyuyoruz ve bu üç tabloyu **join** deyimiyle birleştiriyoruz.Burada Ay ve Yıl olarak tarihsel farkı **DATEDIFF** deyimi ile hesaplıyoruz ve **Where** koşulunda buradaki farkların 0 olması koşulunu arıyoruz. Bu iki koşulun aynı anda olması için de **AND** deyimini kullanıyoruz.

5.Triggerların Etkileri ve Sonuçları

Bu veritabanı için iki tane Trigger tanımladım.

5.1 trg_DeleteUser Triggerı

```
CREATE TRIGGER [dbo].[trg_DeleteUser]
ON [dbo].[Users]
AFTER DELETE
AS
BEGIN
    DELETE FROM PurchasedGames
    WHERE UserID IN (SELECT UserID FROM deleted);
END;
GO
```

Bu Trigger **Users** tablosu için tanımlanmıştır. Bu Trigger **Users** tablosundan bir silinen Kullanıcıya ait bilgilerin **PurchasedGames** tablosundan da silinmesini sağlamaktadır. Bu triggerla sitemden kaydını silen bir kullanıcının oyun bilgilerinin de sistemden silinmesi sağlanmaktadır.

Bu Triggerdaki tetiklenme olayı **Users** tablosundan bir veri silinmesine göre tasarlanmıştır. Begin ve End kısımları arasında ise bu silme işlemlerinden sonra ne yapılacağını gösteren sorgu yazılmıştır. Bu senaryoda yazılan sorgu ise **Users** tablosundaki silinmiş UserID'yi seçip **PurchasedGames** tablosunda bu id ile eşleşen kayıtları silmesini sağlamaktadır.

5.2 info_purchaseGame Triggerı

```
CREATE TRIGGER [dbo].[info_purchaseGame]
ON [dbo].[PurchasedGames]
AFTER INSERT
AS
BEGIN
    print 'One of the users purchased a game!'
END;
GO
```

Bu trigger **PruchasedGames** tablosuna ait bir triggerdır **PruchasedGames** tablosuna insert işlemi yapıldıktan sonra kullanıcılardan birinin oyun satın aldığına dair bir ifadeyi konsola yazdırır.**After Insert** deyimi burada bilgi ekleme işleminden sonra olayını meydana getirmektedir.

6.Sonuçlar

Bu veritabanı Projesiyle birlikte ilişkisel bir veritabanı oluşturma, tablolar arasındaki ilişki türlerini belirleme, tablolara çok fazla sayıda veri ekleme, Sürekli sorgulanan sorgular için stored procedure yazma, belirli özellikleri görebilmek için viewlar oluşturma, bir tabloya veri eklerken veya tablodan veri silerken başka aynı zamanda başka bir tabloda veya aynı tabloda başka işler yapmayı sağlayan triggerlar yazmayı, sorgularda join ifadesini kullanarak tabloları birleştirmeyi ve daha karmaşık sorgular yazmayı öğrendim. Bu alandaki konularla ilgili başka çalışmalar da yapabilecek kadar deneyim sahibi oldum.