**1.3. Çalışma Soruları**

**1.** Transaction nedir?

"Transaction", daha küçük parçalara ayrılamayan en küçük işlem yığınına denir. Belirli bir grup işlemin arka arkaya gerçekleşmesine rağmen, işlemlerin seri ya da toplu halde değerlendirilip hepsinin düzgün bir şekilde ele alınması gerektiğinde kullanılır. "Transaction", prensip olarak ya bütün işlemleri gerçekleştirir ya da hiçbirini gerçekleştirmez.  İşlemlerden biri dahi başarısız olursa, bu prensip nedeniyle hiçbir işlem olmamış kabul edilir; ancak tüm işlemler başarılı olduğunda" transaction", içinde gerçekleşen tüm veri değişikliklerini onaylamış demektir.

**2.** Örnek veri tabanı üzerinde Kitaplar ve Kitap\_Yazar tabloları üzerinde değişiklikler yapılmak isteniyor. Bu işlem için nasıl bir transaction tanımlanabilir.

BEGIN TRAN UPDATE **Kitaplar,Kitap\_Yazar**  
SET kitap\_adi = 'isim1'  
WHERE ISBN = ' DE38C434-5D30-4B28-81A8-5B1A191C1501 '  
UPDATE **Kitaplar, Kitap\_Yazar**  
SET kitap\_adi = 'isim2'  
WHERE musterino = '2'  
IF @@ROWCOUNT = 1   
COMMIT TRAN  
ELSE  
ROLLBACK TRAN

**3.** Transaction ile veri kurtarma ve yedekleme /geri yükleme arasında nasıl bir ilişki olduğunu araştırınız.

VTYS yedekleme ve geri yükleme işlemlerini transaction loglan üzerinden gerçekleştirir. Yedekleme birimi düzenli olarak veritabanını ve transaction loglan yedekleme diskine yedekler. Herhangi bir sebepten dolayı mevcut disk zarar gördüğünde veri ve transaction loğlar okunamaz hale gelebilir. Bu durumda yedekleme biriminden yardım alınacaktır. Yedekleme biriminden veritabanının çalışan son hali alınabilir veya transaction logları üzerinden veritabanının son durumuna geri dönülebilir. Bu sayede mevcut disk zarar görmeden önceki son haline geri dönülmüş olacaktır.

**4.** Bir transaction içerisinde savepoint’ler oluşturulur ve transaction işlemi ROLLBACK ile sonlandırılırsa oluşturulan savepoint’lerin bir önemi var mıdır?

Transaction İçerisindeki SQL ifadelerinden herhangi birisinin yapmış olduğu değişiklikleri geri almaktır. Transaction işlemi İçerisindeki SQL ifadelerinden birisi işlevini yerine getiremediği zaman rollback işlemi ile veriler ilk haline geri getirir. İşlemlerin olduğu yerde commit kullanırsanız savepoint’leriniz yok olur. Savepoint’ler rollback yapılması için bulunmaktadır ve commit edilen bir bilgiyi size geri veremezler. Bunu flashback query yapabilir.

**5.** Bir transaction içerisindeki savepoint’ler COMMİT savepoint ile veritabanına yansıtılır ve transaction ROLLBACK ile sonlanırsa yapılan değişikler veritabanına yansıtılır mı?

RollBack metoduna, parametre olarak string türden SavePoint' in adı verilir. Burada önemli olan, RollBack ile herhangi bir SavePoint' e dönülmesinden sonra eğer Commit işlemi uygulanırsa, SavePoint' e kadar yapılan iş parçalarının kesin olarak veri tabanına yansıtılacağıdır

**2.17. Çalışma Soruları**

**1.** Satış fiyatı en yüksek aracın satış fiyatını bularak @max\_sat\_fiy değişkenine atan ve değişkenin içeriğini yazdıran T-SQL ifadesini yazınız.

declare @max\_sat\_fiy int

select @max\_sat\_fiy=MAX(satis.sfiyat) from satis,arac where arac.aracno=satis.aracno

print(CAST(@max\_sat\_fiy as varchar(20)))

**2.** Sistem değişkenleriyle yerel değişkenler arasındaki fark nedir?

Yerel değişkenler herhangi bir türde sadece bir veri tutabilen nesnelerdir. SQL Server Integration Services’da çalışan paketi ve onun nesneleri hakkında bilgi depolayan değişkenler sistem değişkenleridir.

**3.** En son oluşan hatanın kodunu ve açıklamasını yazdırmak için gerekli T-SQL ifadesini yazınız.

select @@ERROR as 'Hata Kodu: ', @@IDENTITY as 'Tanımı: '

**4.** Müşterilerin ad ve soyadlarını musteriler isimli bir tablo değişkenine ekleyerek tablo değişkeninin içeriğini yazdıran T-SQL ifadesini yazınız.

declare @musteri table (ad varchar(25),soyad varchar(25))

insert into @musteri values ('Turgut','Özseven')

insert into @musteri values ('Mustafa','Çağlayan')

insert into @musteri values ('Ahmet','Kara')

insert into @musteri values ('Murat','Beyaz')

insert into @musteri values ('Elif','Kurt')

insert into @musteri values ('Ayşe','Uçar')

insert into @musteri values ('Bülent','Ayar')

select \* from @musteri

**5.** 3 nolu araçtan yapılan satışlar var ise listeleyecek yok ise ‘Bu araçtan satış yapılmamıştır.’ Mesajını yazacak T-SQL ifadesini yazınız.

select \* from satis where satis.aracno=3

if(@@ROWCOUNT>0)

begin

select musteri.madi,musteri.msoyadi,arac.marka,satis.strh,satis.sfiyat from musteri,arac,satis where satis.aracno=arac.aracno and satis.mno=musteri.mno and satis.aracno=3

end

else

begin

print('Bu araçtan satış yapılmamıştır.')

end

**6.** Araç tablosunda bulunan araçlardan en yüksek modele sahip aracın fiyatı 34000’den fazla olana kadar araçların fiyatlarını %10 arttıran T-SQL ifadesini yazınız.

declare @fy money

select @fy=arac.fiyat from arac where arac.model in (select MAX(arac.model) from arac)

while(@fy<=34000)

begin

update arac set arac.fiyat+=arac.fiyat\*0.1

select @fy=arac.fiyat from arac where arac.model in (select MAX(arac.model) from arac)

end

select \* from arac

**7.** aracbul isminde bir yordam oluşturunuz. Yordam dışarıdan araç numarasını alacak ve bu numaraya sahip aracın alımı veya satışı yapılan araçların markasını bulacaktır.

create function aracbul(@nn int=0)

returns @geri table(marka varchar(25))

as

begin

insert into @geri (marka)

select arac.marka from arac,alim,satis where arac.aracno=@nn and (arac.aracno=satis.aracno or arac.aracno=alim.aracno) group by arac.marka

return

end

select \* from aracbul(2)

**8.** satis\_yap isminde bir yordam oluşturunuz. Yordam dışarıdan müşteri no ve araç no parametrelerini alacak ve gelen bilgileri satış tablosuna ekleyecektir. Satış tarihi olarak o anki tarih kullanılacaktır.

create function satis\_yap(@mno int, @ano int, @fyt money)

returns @tt table(mno int,aracno int,strh date,sfiyat money)

as

begin

insert into @tt values(@mno,@ano,GETDATE(),@fyt)

return

end

insert into satis(mno,aracno,strh,sfiyat) select \* from satis\_yap(2,3,5000)

**9.** Parametre olarak aldığı araç numarasına ait aracın markasını bulan fonksiyonu oluşturunuz. Oluşturulan fonksiyonu kullanarak satışı yapılan araçların markasını bulunuz.

create function ab(@nn int)

returns @geri table(marka varchar(25),an int)

as

begin

insert into @geri (marka,an)

select arac.marka,arac.aracno from arac,satis where satis.aracno=arac.aracno and arac.aracno=@nn

return

end

declare @i int,@j int

set @i=0

select arac.aracno from arac

set @j=@@ROWCOUNT

while(@i<=@j)

begin

select \* from ab(@i)

set @i+=1

end

**10.** İmleçlerin görevini ve türlerini açıklayınız.

İmleçler, SQL Server üzerinden döndürülen belli bir kayıt kümesi üzerinde satır satır işlem yapabilmeye olanak sağlayan veritabanı nesneleridir.

-Statik İmleçler

-Dinamik İmleçler

-Salt ileri İmleçler

-Tuş Takımıyla Denetlenen İmleçleri

**11.** İmleç kullanarak müşteriler tablosunun içeriğini listeleyiniz.

DECLARE @adi VARCHAR(50),@sadi varchar(25),@adres varchar(70),@tel varchar(20)

DECLARE musteriler CURSOR FOR

SELECT \* from musteri order by madi

open musteriler

FETCH NEXT FROM musteriler INTO @adi,@sadi,@adres,@tel

WHILE @@FETCH\_STATUS =0

BEGIN

SELECT \* from musteri

FETCH NEXT FROM musteriler INTO @adi,@sadi,@adres,@tel

END

**12.** Satışı yapılan herhangi bir aracın satış fiyatı araç tablosunda bulunan fiyattan yüksek ise araç tablosundaki fiyat bilgisini yeni fiyat bilgisi ile değiştirecek tetilleyici oluşturunuz. Örneğin, Ford Focus aracının fiyatı 28.000’dir. Eğer bu araç 29.000’dan satılırsa araç tablosundaki bu aracın fiyatı da 29.000 olarak değiştirilecektir.

Create trigger sts

on satis

for insert

as

begin

declare @yfyt money,@ano int,@fyt money

select @ano=aracno,@yfyt=sfiyat from inserted

select @fyt=fiyat from arac where arac.aracno=@ano

if(@yfyt>@fyt)

begin

update arac set arac.fiyat=@yfyt where arac.aracno=@ano

end

end

insert into satis values(6,7,4,'15.06.2010',30000)