--Lutfen calısanlar tablosuna SilindiMİ SUTUNU EKLEYINIZ

alter table Employees add SilindiMİ bit

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

--birisi calısanlar tablosunda veri silmek istediginde silindimi sutunu otomatık olarak 1 olsun,silme işlemi yapılmasın.

CREATE trigger PersonelSilme

on Employees

instead of delete

as

update Employees set SilindiMi=1 WHERE EmployeeID=(SELECT EmployeeID FROM deleted)

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

DELETE FROM Employees WHERE EmployeeID=1 --silme işlemi deilde onun yerıne null yazan yere true yazıyor.

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

--nakliyeciler tablosunda yapılan tum degisiklikleri notlar tablosunda saklayınız.

alter trigger NakliyeciDinleme

on Shippers

for update,insert,delete

as

if exists(SELECT \* FROM inserted) AND exists(SELECT \* FROM deleted)

begin

insert into Notlar (NotAciklama) values ((SELECT CompanyName FROM deleted)+'isimli sirket guncellendi')

end

else if exists(SELECT \* FROM inserted)

begin

insert into Notlar (NotAciklama) values ((SELECT CompanyName FROM inserted)+'isimli sirket yeni eklendi')

end

else if exists(SELECT \* FROM deleted)

begin

insert into Notlar (NotAciklama) values ((SELECT CompanyName FROM deleted)+'isimli yeni silindi')

end

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

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

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

------------------------------------HATA KONTROLU----------------------------------------------------------------------

--1 STOK GİRİSİ

--2 KİRALAMA

--3 KİRA DONUSU

CREATE TABLE StokHareket

(

HareketNo int primary key identity(1,1),

CdNo int,

HareketTanim int

)

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

CREATE PROC HareketGirisi

@cdNo int,

@hareketTanim int

as

if(@hareketTanim = 2) --kiralama islemi ise

begin

declare @sonHareket int =

(SELECT top 1 @hareketTanim from StokHareket where CdNo = @cdNo order by HareketNo desc)

if(@sonHareket = 1 or @sonHareket = 3)

begin

insert into StokHareket (CdNo , HareketTanim) values (@cdNo,@hareketTanim)

end

else

begin

raiserror('bu cd kiraya cıkamaz.Kira donusu yapilmamis yada stoga hic girmemis',16,1)

end

end

else

begin

insert into StokHareket (CdNo,HareketTanim) values (@cdNo,@hareketTanim)

end

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

exec HareketGirisi 3,2

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

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

SELECT CONVERT(int,'12a')

--begin try

--hata vermesi muhtemek kodlar bu blok ıcıne yazılır

--end try

--begin catch

--eger bir hata cıktıysa ne yapılması gerektigini yazılır.

--end catch

--level 10 kadar ciddi hata varsaymıyor buyuzden catch ı calıstırmaz

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

declare @sayi int

begin try

select @sayi=convert(int,'12a')

end try

begin catch

set @sayi=25

end catch

select @sayi

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

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

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

------------------------------------------------TRANSACTION----------------------------------------------

--birer nesne degıllerdır(trigger view gibi degıl) create ıle yaratılmak yerıne dogrudan kullanılıyorlar.

--calıstırılan bırden fazla sorgunun tek bır sorgu olarak algılanması ıcın kullanılır.

CREATE database HalkBankasi

go

use HalkBankasi

go

create table Hesaplar

(

HesapNo int primary key identity(1,1),

HesapAdi varchar (50),

Bakiye decimal(8,2)

)

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

create proc ParaCek

@HesapAdi varchar(50),

@Miktar decimal(8,2)

as

update Hesaplar set Bakiye=Bakiye-@Miktar where HesapAdi=@HesapAdi

go

create proc ParaYatir

@HesapAdi varchar(50),

@Miktar decimal(8,2)

as

update Hesaplar set Bakiye=Bakiye+@Miktar where HesapAdi=@HesapAdi

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

go

declare @Miktar decimal(8,2) =100000

exec ParaCek 'Aydın',@Miktar

exec ParaYatir 'Yonca',@Miktar

go

declare @Miktar decimal(8,2) =400000

exec ParaCek 'Yonca',@Miktar

exec ParaYatir 'Fırat',@Miktar

--burda sorun cıkıyor cunku decımal 8,2 belirlemistik ama fıratın bakıyesı 8 haneyi gectıgı ıcın ,aslında yoncanın hesabından para cekılır fakat fırat hesabına aktarılmıs gorunmez.bu cok sıkıntılı bır durumdur.bunu ıcın transaction kullanmalıyız.buyuzden;

go

begin try

begin tran

declare @Miktar decimal(8,2) =400000

exec ParaCek 'Yonca',@Miktar

exec ParaYatir 'Fırat',@Miktar

commit tran -- kod calısma esnasında declare exec basarılıyla gecılmısse onay olarak commıt tran kullanılır.

end try

begin catch

--eger yukarıdakı kısım dogru cıkmazsa catch kısmı calısır.

rollback tran --yapılan ıslemlerı gerı sar anlamındadır.eger bunu demezsek yoncanın parası eksiktir.

end catch

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

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

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

USE Northwind

GO

begin tran

SELECT ContactName FROM Customers WHERE CustomerID='ALFKI'

update Customers set ContactName ='Meryem Andır'

SELECT ContactName FROM Customers WHERE CustomerID='ALFKI'

rollback tran

SELECT ContactName FROM Customers WHERE CustomerID='ALFKI'

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

----------------INDEX-------------------------------------

--2 cesit index vardir

declare @index int = 0

while(@index <= 1000000)

begin

insert into Shippers (CompanyName,Phone) values ('Sirket'+ convert(varchar,@index),'3366999')

set @index +=1

end

--table scan:heap teki her bir verıye ulasmak ıcın yapılan ıslemın aynısı.

--clustered index:kitapları alfabetik sıraya koyma islemidir.

--bir kitap sadece bir ozellıgıne gore sıralanabilir.

--non-clustered index:sadece kıtabın nerde oldugunu bır lısteleme ıslemıyle buluyoruz.

--Bir tabloda en fazla 249 tane nonclustered index tanımlayabiliriz.

-------------Query Optimizer-------

--ilgili kayıt ıcın ne tur yontem kullanacagını,yani kayıda ulasmak ıcın kar zarar yaparak bilgiye en kısa en verımlı nasıl ulaşacagımıza bakar.

--query optimizer in bazı terımlerı mevcut bunlar;

--extend:64 kb lık extend denilen alanlara böler.verileri tutmak icin ayrılmıs ham alanlardır.USB bellege benzetilebilir.

--asıl verının tutuldugu 8 kb lık page denılen mantıksal yapılardan olusur.yani extend,8 tane 8 kb lık page den meydana gelır.Page ler ise verilerimizin bir satır duzenındedır.bunlarda rows olarak nıtelendırılır.

--bir index en fazla 16 kolon ıcerır.

--Rebuild:indexin kullandıgı alanı yenıden duzenlemek ıcın ındexi silip yenıden olusturma ıslemıdır.Amac: kapladıgı alanı azaltmak.(online offlıne yapılabılıyor)

--Reorganize:leaf levelı ındex lemek yenıden duzenlemek ıcın yapılır.dıger sevıyelerde duzenleme yapılmaz.

--index lenmeyen veri tipleri; varchar(max),nvarchar(max),image,xml,ntext,text

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

--MANTIKSAL OKUMALARI ıo olcer

--kac mantıksal okumaya yaptıgını ıo ıle bu ıolayın kac sn aldıgını tıme on ıle olcucez.

set statistics IO on

set statistics time on

SELECT \* FROM Shippers WHERE CompanyName='Sirket684723'

-- CPU time = 266 ms, elapsed time = 316 ms.

-- logical reads 7147

--SQL Server Execution Times: CPU time = 218 ms, elapsed time = 647 ms.

--index li hali asagıdaki gibidir:

CREATE index indNakliyeci on Shippers(CompanyName) --indexleme yapıldı

--sunucu restart edilir.

set statistics IO on

set statistics time on

--tekrar calıstırılır

SELECT \* FROM Shippers WHERE CompanyName='Sirket684723'

--CPU time = 172 ms, elapsed time = 787 ms.

--logical reads 6

-- CPU time = 0 ms, elapsed time = 53 ms.

--verılerden yola cıkarak; mantıksal okuma sayısı ındexlenınce cok fazla dustugu goruluyor.

--non clustered ındexleme yaptık.

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

--rebuild:index i kaldırıp yenıden olusturur.

alter index indNakliyeci on Shippers rebuild

--reorganize:bozulmus data page leri duzeltır.(leaf tabloları yeniden olusturur.)

alter index indNakliyeci on Shippers reorganize

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

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

--Proseduru functıon gıbı kullanmak--------------------------

exec sp\_serveroption 'PC','DATA ACCESS',TRUE --İZİN ALIYORUZ

--BUNU YAZDIKTAN SONRA HERHANGI BIR PROSEDUR ICINE SELECT CEKMELIYIZ

CREATE proc FiyatiGuncelle

as

UPDATE Products set UnitPrice=UnitPrice+5 WHERE ProductID in(

select ProductID from OPENQUERY(PC,'exec Northwind.dbo.UrunVerStokIle 20 '))

--uzunca bole yazılır