**SORU 1: Bir veri tabanında beklediğimiz nitelikler nelerdir?**

Öncelikle veri tabanı(database) nedir? Belirlenen özel amaçlar doğrultusunda birbirleriyle ilişkili verilerin belirli kurallar çerçevesinde bir araya getirilmesiyle oluşturulan koleksiyona veri tabanı (database) denir. Bizlere program yazılması için gereken zaman harcanmasını ortadan kaldırır. Kendisi(veri tabanı) zaten önemli fonksiyonları sağlar.

Bir veri tabanı veri tabanına kaydedilecek veriyi en uygun bir biçimde modellemeyi, veri tabanı denetim, erişim ve yönetim metotlarını uygun bir biçimde kullanarak çeşitli uygulamaları geliştirmeyi hedefler.

ER(Varlık-İlişki) modeli, UML(Unified Modeling Languege), ilişkisel Model, ilişkisel cebir, SQL(Structured Query Language), Dosya Yapıları ve İndexleme, Sorgu Optimizasyonu, Hareket Yönetimi(Transactions), Kilitlenmeler(Deadlocks), Eşzamanlılık(concurency), ve Veri tabanı Kurtarma Teknikleri(Recovery) veri tabanı yönetim sistemlerinin konularındandır.

Veri tabanında beklenilen nitelikler; başarılı bir veri tabanları uzun yıllar boyu kullanılacağından veri tabanı gelecekte istenilecek olan değişimlere(changebilility) ve gelişimlere(evolvability) cevap verebilecek biçimde tasarlanmalıdır.

Veri tabanında ki Sofistike ve teknik bilgisi olmayan kullanıcıların sorgularını kullanma grubuna göre yönlendirilmesini veya yönetmesini görürürüz.

Veri tabanı veri modeli tabanlı sınıflandırma yapılmalı. (hiyerarşik-ağ modeli-nesne tabanlı-nesne ilişkisel-ilişkisel veri modeli) bunlar normalleştirme ile oluşturulup veri modelleme sürecinde normalizasyona bakılır. Tablo yönelimli ilişki şeması varlıklara karşılık gelir. N tane harici şema olabilir.Harici şema ile verinin farklı alanları ve görünüşleri(view) sunulur. Kavramsal şemalar bütün ilişkileri tanımlar.(Varlıklar ve varlıklar arasındaki ilişkiler). Kullanıcıların kayıt seviyesindeki değişikliklerden etkilenmelerini minimum seviyeye indirir. Fiziksel şema ile bunların nasıl kaydedileceğinin bilgilerini içerir. Index oluşturur veriye erişimi birinci dereceden etkilemektedir.

Veri koleksiyonundan istenilen özellikler aşağıdakiler olabilir;

-Birden fazla kullanıcının (n) eş zamanlı olarak erişiminin sağlanması

-Yeni veri eklenebilmesi

-Var olan kayıtların güncellenebilmesi

-İstenilen kayıtların silinebilmesi(Nadir kullanılabilir)

-Sorguların makul süre içerisinde cevaplandırılması

-Verinin istenilen kısımlarına erişim kısıtlaması getirilmesi. Yani farklı erişim seviyeleri tanımlamaya imkan sağlanır

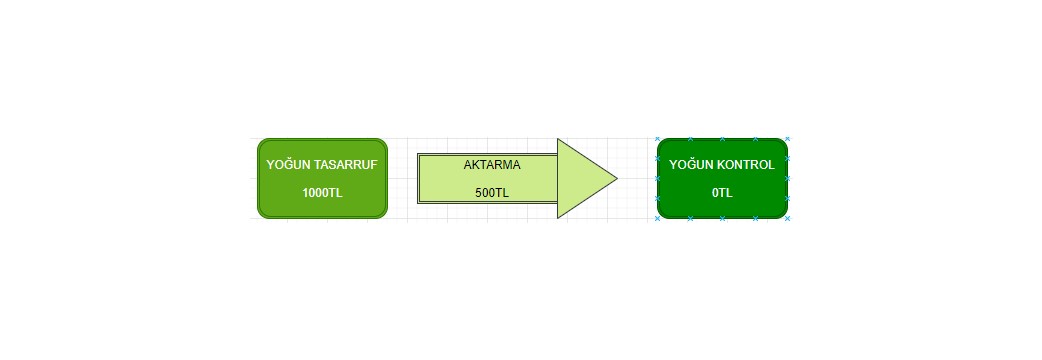
Ek olarak birbirinden bağımsız olarak işlem gerçekleştirebilme imkanı sağlamasını veri tabanı katmanlarından bekleyebiliriz. Genel görünüşü verilip detayları izole etmesini bekleyebiliriz. Veri ekleme, güncelleme sonra tekrar okuma ve elde etme(erişim) işlemleri etkili veri bağlantılarında beklenilen tekniklerdendir. Birden fazla değişim isteğinin kısıtlanması farklı kullanıcılar için erişim denetimi sağlanıp buradan veri tabanı üzerinden erişilen verinin bütünlüğünün korunması ve bu şekilde güvene alınması beklenir.

Açıklama olarak; Veri yönetimi sağlanarak tasarlama ve düzenleme sonucunda veri sunumunda tekrarlamalar ortadan kaldırılarak verinin çağrılması (sorgusu) optimize edilir. Yukarıdaki maddelerde yer alan eş-zamanlı erişimlerde her seferinde bir kullanıcının aynı veriye erişimini düzenler sistem hatalarından ve etkilerinden kullanıcıları ve verileri korur.

**SORU 2: Transaction kavramı nedir? Hangi durumlarda kullanılır? Örnekler ile detaylı biçimde açıklayınız.**

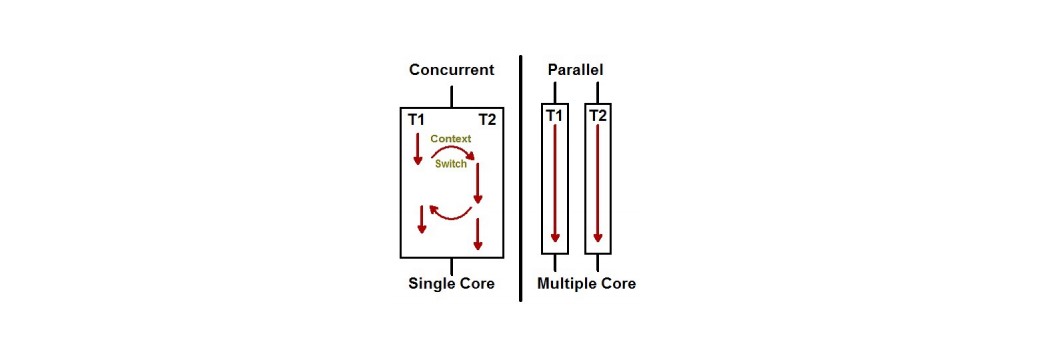
Bir kullanıcının veritabanı yönetim sistemleri içerisinde gerçekleştirmiş olduğu CRUD işlemlerinin(execution) her birine ve bu crud işlemlerinin içerisindeki her bir işleme (create-read-update-delete) **hareket(Transaction)** adını veriyoruz.

Bir veri tabanında bir hesaptan başka bir hesaba para aktarımı söz konusuysa hareketler bütünü olarak ifade edilir. Bunun için gerçekleşenler; göndericinin hesabı kontol ediliyor, gönderilmek istenen miktar mevcut ve işlem için yeterli ise hesaptan düşürülüyor, sonrasında gödericinin banka hesabı güncelleniyor yeni miktarı ile, alıcı hesabı kontrol ediliyor, alıcı hesabına istenilen mktar gönderiliyor ve alıcı hesabı güncelleniyor.



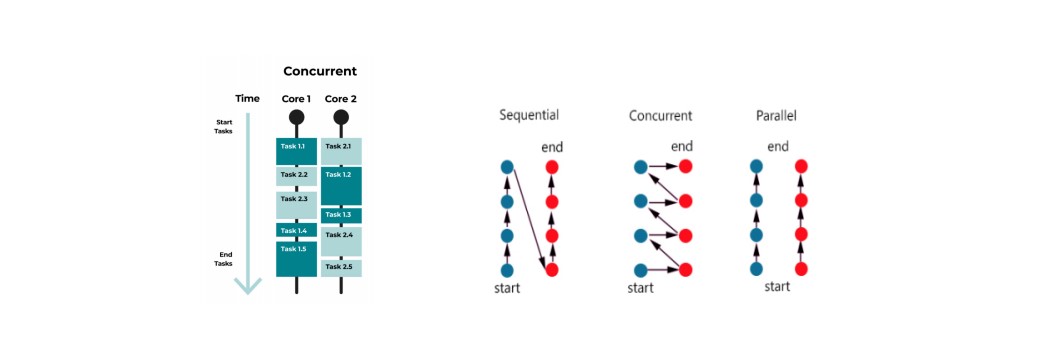
Müşteri açısından tek bir işlemBirkaç işlemden oluşur.Tüm bu işlemler gerçekleşmeli veya hiçbiri gerçekleşmemelidir.

Hareket yönetimi gerçekleştirilirken veri tabanı yönetim sistemlerinin kullandığı önemli yöntemlerden biri **Concurrency**’dir. Bu eşzamanlılık olarak ifade edilir. Özellikle sistem hatalarından korunmanın temelini oluşturmaktadır. Eş zamanlı veri işleme ile paralel biririne karıştırılabiliyor.



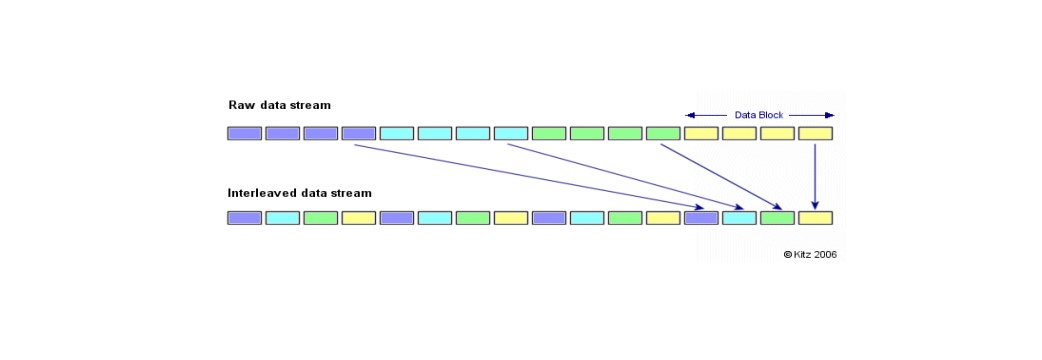
Bizlerin bilgisayarında bulunan her bir CPU’nun kendi içerisinde gerçekleştirebileceği işlem dizisinin belli olduğu ve birden fazla transactionun değilde tek bir hareketi gerçekleştirdiğini söyleyebiliriz. Yahut bir görevi bir CPU’nur core’na vermiş olduğu bir görevi ve bunu tamamlaması paralel olarak gerçekleştirilir ama eşzamanlılıkta tek bir core’da birden fazla hareketin dönüşümlü olarak çalışmasına concurrent eşzamanlılık adını veriyoruz.

Bir problem çözülebilecek ve aynı zaman diliminde hesaplanabilecek alt problemlere bölüp sonucu değişmeyecek bir biçimde belirli bir sıraya uyulmaksızın gerçekleştirilmesine eş zamanlılık denir.



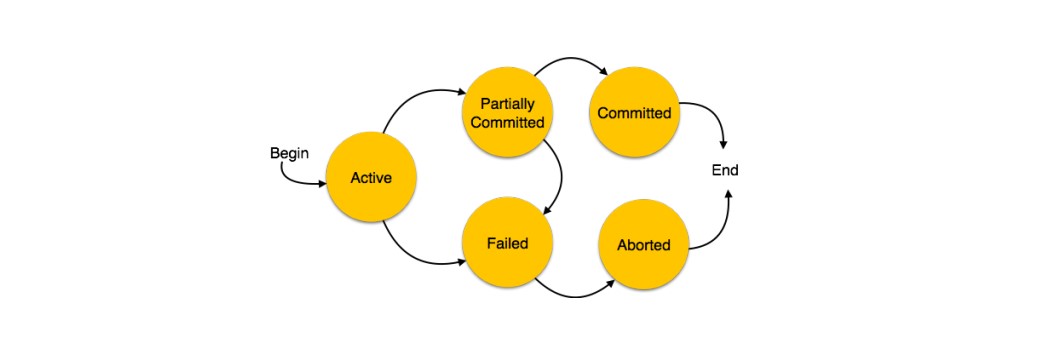
Serilerde sırayla işlem yapılır eşzamanlı gerçekleştirilen işlemler farklı core lar tarafından gerçekleştirilir ama paralelde bu söz konusu değildir.

İstenilen performası gerçekleştirmek için hareketler arasında interlaving (dönüşümlü çalıştırmak) tekniğini kullanmaktadır.



Mor yani ilk dörtlüyü bir görev olarak değerlendirebiliriz. Seride ikinci mavi olan dörtlüye geçilir sonra diğerine en son sarı olan dörtlüye geçilip sırayla işlemi tamamlarız. Bunların Her birini farklı kullanıcıların gerçekleştireceği işlemler olarak düşünün. Her bir kullanıcının problemi küçük alt kısımlara bölünüp ve dönüşümlü olarak çalıştırılıp kullanıcıları beklemeden parça parça her görevi işleyebiliriz.

Aşağıdaki tabloda bir hareket başlar aktif olabilir bu hareket başladıktan sonra direk hata verebilir ve iptal edilebilir ve işlem biter ya da aktif olan işlemin bir kısmı tamamlanır ve daha sonrasında bir kısmı tamamlandıktan sonra hata verir iptal edilir ve yine işlem biter ya da aktif olan bir işlem Partially commited(bir kısmı tamamlanır) ve daha sonra işlemin geriye kalan kısımları tamamlanır (işlemin hepsi) hata vermeden tamamlanır ve daha sonrasında işlem biter.



hareketlerin dört temel özelliği bulunmaktadır. **A**tomicity (bölünemezlik) , **C**onsistency (tutarlılık), **İ**solation (yalıtım) , **D**urability (dayanıklılık)

**Anatomicity:**Bir hareketin tamamlanabilmesi için kümelenmiş sıralı işlemlerin bölünmezliğidir. İşleminlerin sadece bir kısmı gerçekleştirilmez yani hepsi başlar biter veya işlem gerçekleştirilmez. Tamamlanamayan veya yarıda kalan hareketlerin işlemleri veri tabanı tarafından geri alınır.

Mehmet’in hesabında 400tl bulunmaktadır Kamil’in hesabında 700tl bulunmaktadır. Kamil’in Mehmet’e 200 tl gönderilmesi gerekmektedir. Bu transfer iki aşamada gerçekleşmektedir.

1. Kamil’in hesabından 200tl düşülecek 2. Mehmet’in hesabına 200 tl eklenecek

1. Başarılı 2. başarısısz olursa; Mehmet’in hesabında (400tl) ve Kamil’in hesabında ise (500tl) olacak.

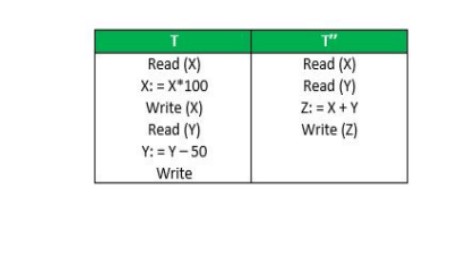
**Consistency (tutarlılık):** Veritabanı gerçekleştirilen hareketlerde kayıtlı bulunan verinin tutarlılığının korunmasına consistency denir.Veritabanı gerçekleştirilen her işlemde tutarlılığın sağlandığını varsaymaktadır.Ilişkisel bütünlük (relational integrity) korunmaktadır. Mesela;

Hesaplar arasında transfer yapılırken toplam miktar aynı kalmalıdır

Hareketten önce Kamil (700)+ Mehmet (400) = 1100

Hareket tamamlandıktan sonra Kamil (500)+ Mehmet (600) = 1100

**İsolation (yalıtım):** Veritabanı yönetim sistemlerinde birden fazla hareket yönetilirken bir hareketin diğer bir hareketten izole edilmesine yalıtım denir.



Bir hesaptan aynı anda (T1 ve T2) para çekildiğinde önce başlanılan işlem (T1) tamamlanmadan diğerinin (T2) tamamlanması beklenir.

**Durability (dayanıklılık):** Sistem hataları ortaya çıktığında veya hareketlerde problemler ile karşılaşıldığında hareket bilgilerini ve kayıtlı olan veriyi kaybetmeden işleme devam edebilmesi durability olarak tanımlanır.

Bir hareket içerisindeki bütün işlemler başarıyla tamamlanmışsa bu işlem commited (teslim etmek) olarak adlandırılır ve yapılan bütün değişiklikler diske kaydedilir.

Bir hareket içerisinde herhangi bir hata oluştuğunda yapılan bütün değişiklikler geri alınarak hareket başlamadan önceki duruma geri dönülmesine rollback (geri dönüş) denir.

Birden fazla işlemin aynı zaman dilimi içerisinde gerçekleştirilmesine concurrent execution (eşzamanlı çalışma) denir.

Belirlenen bir zaman dilimi içerisinde gerçekleştirilen ortalama işlem sayısına throughput (üretilen iş) denir

Aynı zaman dilimi içerisinde gerçekleştirilen işlem performansını arttırmak için concurrency kullanılır.

Sıralı işlemlerde (concurrent olmayan) kısa süreli bir hareket uzun zaman alan bir transaction’ın ardında ise ne zaman biteceği tam olarak öngörülemeyebilir.

Hesap transaction örnek kodu;

BEGIN;

UPDATE accounts SET balance = balance - 100.00 WHERE name = 'Alice';

SAVEPOINT my\_savepoint;

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Bob';

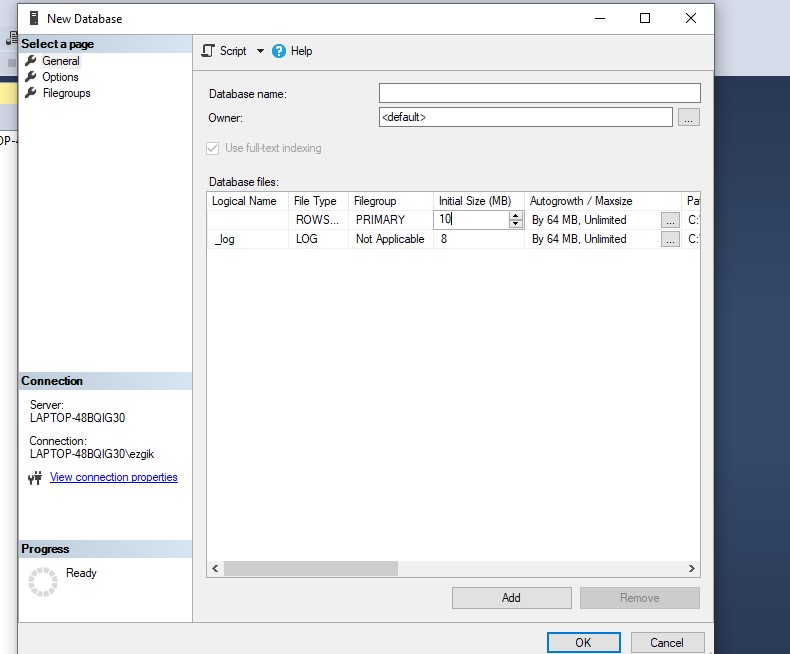
-- oops ... forget that and use Wally's account

ROLLBACK TO my\_savepoint;

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Wally';

COMMIT;

**SORU 3: A) SQL veri tabanına bağlanırken başlangıçta veri tabanının boyutunu belirleleyebilir miyiz?**



10 ile gösterilen yer bize başlangıçta belirleyeceğimiz veri tabanı boyutunu gösterir (install size)

Başlangıçta yazdı yazdı 10Mb’ı doldurdu dolduktan sonra bu veri tabanının büyümesi lazım.Ama neye göre büyüyecek? Iki seçeneğimiz var in percent (yüzdelik olarak) ya da in megabyte (belli bir boyutta) büyütebiliriz. Bu bizim kullandğımız algoritmaya göre değişir.

Veri tabanının kapladığı alanı temel dosya bilgileriyle belirli komut sorgularıyla ve belirli bölüm başlıklarında veri tabanının kapsadığı alan ölçüsünü(boyutunu) sutunlara-kolonlara ait açtığımız alanları belirledikten sonra öğrenebiliriz. Yani bu demek oluyor ki; Boyut dediğimiz depolama alanı depolanacak veri miktarı maksimum boyutuyla sınırlıdır. Bu maksimum sınırlandırma boyutu 1GB(1073741824BAYT) ve katlarından biri olmalıdır diye belirtilebilr.

Kendi içlerinde farklı veri tabanları baştan otomatik olarak bir alan ayırabilir. Boyutlar kendi içlerinde konularına göre atanır. Bildiğiniz üzere tarih-zaman, parasal, sayısal, metinsel, binary(ikilik) ve diğer veri tipleri boyutları mevcuttur yani bir verinin tipine göre değişiklik gösterir ama bu tipi belirli tiplere kendi isteğimize göre uygulayıp kendimiz ölçeklendirebiliyoruz.

Örnek olarak;parasal bir miktar belirteceğim int(integer) olan bir veri tipini yerine isteğimi kapsayacak ölçekle bigint olan değer olarak belirtebilirim. Ikiside sonuç olarak sayısal veri tiplerine girmekte ama int’in 4bayt begint’in 8bayt büyüklüğünde bir tam sayı veri tipi olduğunu unutmamalıyız.

Bunu sql formatında oluşturacağımız bir veri tabanı tablosu kodunu script halinde küçük bir örnek ile göstereck olursak;

Create database okul

Create table ögrenciler

(

ögrno int identity(1,1) primary key,

Adi\_soyadi varchar (30),

Yasi integer ,

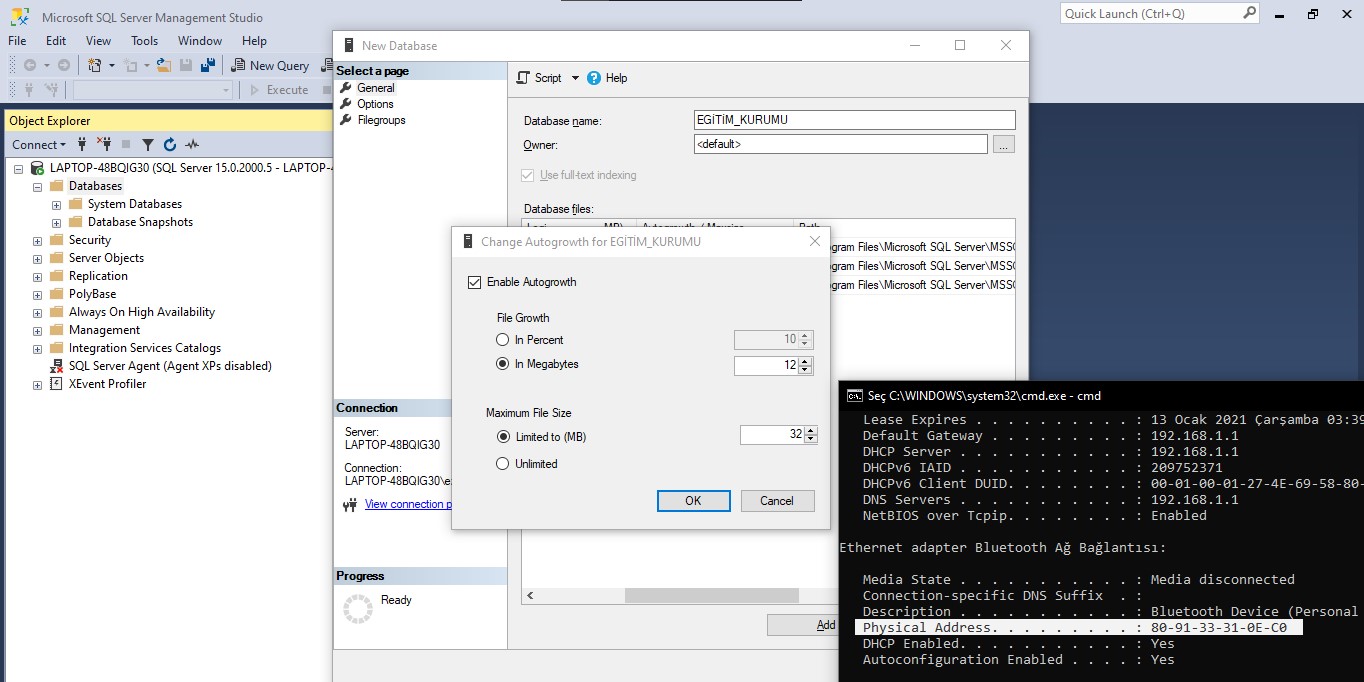
Adres text

)

Burada oluşturduğumuz okul isimli veri tabanımızın öğrenciler tablosuna veri tipleri değeri olarak varchar, integer, text tipleri olarak belirttim. Adresi varchar olarak atayıp istediğim en uzun karakter sayısına kadar adi\_soyadi kısımdaki gibi belirtebilirdi ama text yazılıp girilen dğer kadar boyut kaplamasını tercih ettim. Kısacası SQL veritabanına bağlanırken gördüğünüz üzere başlangıçta veri tabanı boyutunu belirleyebiliyoruz.

**B) Tuttuğu veri boyutu arttıkça otomatik artış sağlayabilir miyiz? Eğer yapabiliyorsak hangi özellikleri ile yaparız? Yapabiliyorsak yeni bir veri tabanı oluştururken bu ayarları yaptığımız pencerede başlangıç boyutu 32MB, artış miktarı 12MB olan bir veri tabanı oluşturma esnasındaki ekran görüntüsünü bu sorunun cevabı olarak gösteriniz.**

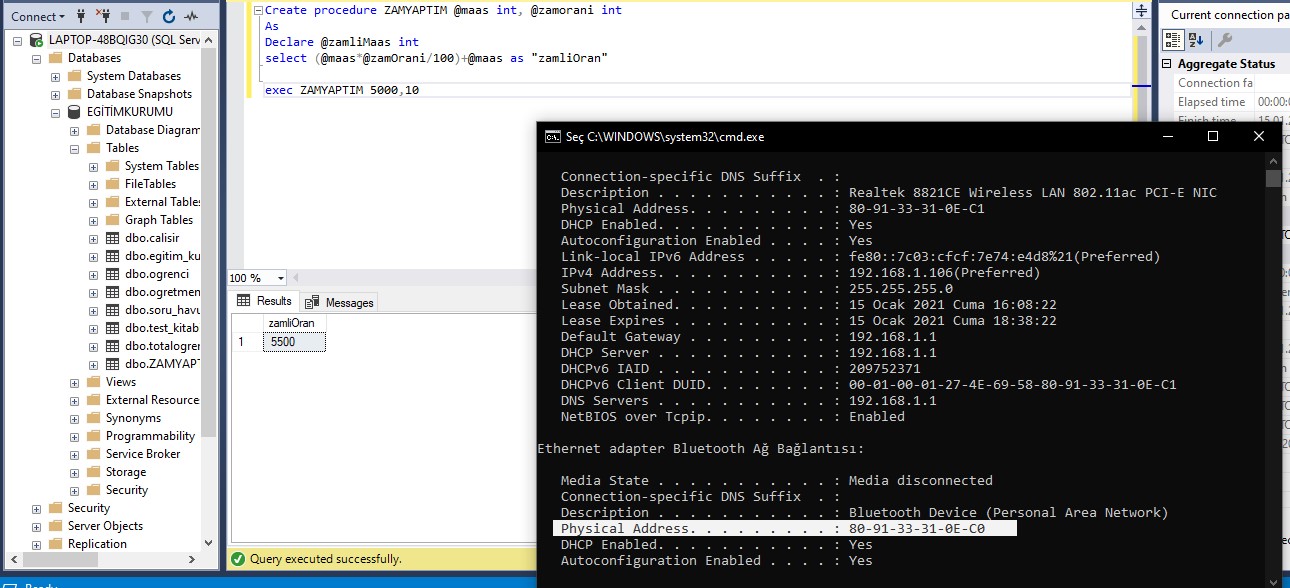
Evet otomatik artış sağlanabilir.



**Physical address: 80-91-33-31-0E-C0**

Örnek olarak; Veritabanı ve log dosyası C:\ sürücüsünde belirtilen ve boyutlarda oluşturulacaktır. Veritabanı dosyası maksimum 50 MB, başlangıç boyutu 10 MB ve log dosyası maksimum 25 MB, başlangıç boyutu 5 MB olarak ayarlanacaktır. Her iki dosyada belirtilen değere ulaştığında boyutu 5 MB artacaktır

**4)Dışarıdan ‘maaş’ ve ‘zam oranı’ isminde iki parametre alan ve mevcut maaşı zam oranında arttıran ‘ZAM YAPTIM’ isminde stored prosedüre oluşturunuz.**



Create procedure ZAMYAPTIM @maas int, @zamorani int

As

Declare @zamliMaas int

select (@maas\*@zamOrani/100)+@maas as "zamliOran"

exec ZAMYAPTIM 5000,10

**5) Tasarlayıp kodladığınız bir veri tabanında güncelleme sorgusu çalıştırılacağı zaman güncelleme işlemi yapılmadan önce mevcut bilgilerin taşınabilir bir bellekte depolanmasını sağlayacak bir yöntem bulmanız istenmektedir. Bu işlemi gerçekleştirebilmek için izleyeceğinin yöntem/yöntemler nelerdir? Teorik olarak izleyeceğiniz adımları anlatınız. (izleyeceğiniz yöntem/yöntemler SQL’in desteklediği tetikleyici , yapı, yordamlardan biri olmak zorundadır.)**

Elle veya uygulamalar yardımıyla yedeklerimizi yapabilir ama pahalı maliyette olanı tabii ki seçmek istemeyiz.

Depolama yapılarında sistem çökmelerinden kurtulamaz örnekler ; ana bellek ,ön bellek. Kalıcı olmayan depolama sistem çökmelerinden kurtulur örnekler; disk, teyp, flash bellek , uçucu olmayan (pil yedeklemeli) RAM. Astable depolamada tüm arızalardan kurtulan efsanevi bir depolama biçimidir. Farklı kalıcı olmayanlar üzerinde birden çok kopya muhafaza edilerek yaklaşık medya yeri kaplar.

*Sunucu da içerik kapsamlı DDL tetikleyicileri, database veritabanında nesneler olarak konumlandırılır.**Veri-database kapsamlı DDL tetikleyicileri, oluşturuldukları veritabanında nesneler olarak saklanır.* yedekleme dosyasında RESTORE HEADERONLY komutunu çalıştırın. Bir yedekleme dosyası birden fazla yedekleme içerebilir ve DOSYA İLE seçeneğini belirtmediğiniz sürece dosyadaki en eski yedeklemeyi alırsınız.

Depolama(yedekleme) den önce Backup seçeneklerini bilmek gerekir full backup(tam yedek),diferantial backup(sadece değişkenlerin yedeği),transaction log backup(işlem bilgilerinin tutulduğu) beckuplardır.

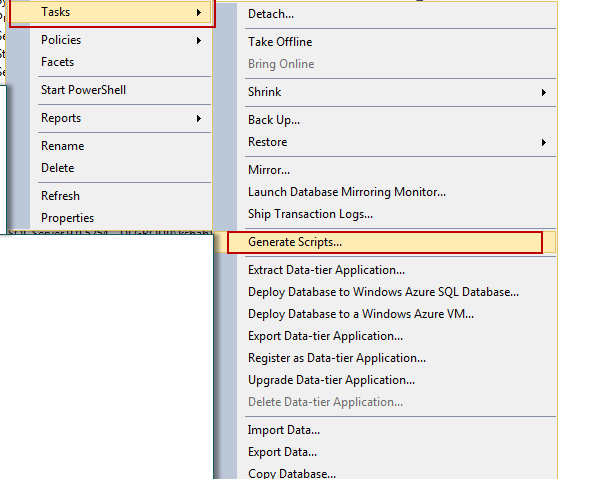
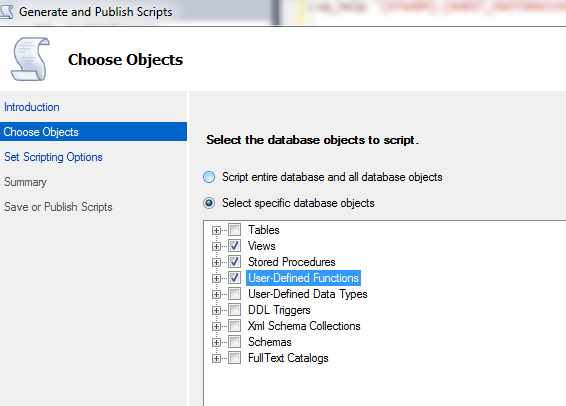
Recovery model yapılandırılması bunlardan önce yapılması gerekendir. Database de sağa tıklanıp properties’e tıklanır. Options sekmesinden yapmak istediğimiz backup ayarlaması yapılır. Bunları yaptıktan sonra database iki farkliı şekilde alınabilir . Bu .bak dosyası ve backup device ile. .bak’da databasenin sağına tıklanıp tasks seçilir ve backup seçeneğine tıklanır. Burada da destination, backup compenant ve source bölümleri bulunur. Bunlar full , simple ve bullk loged’dir.

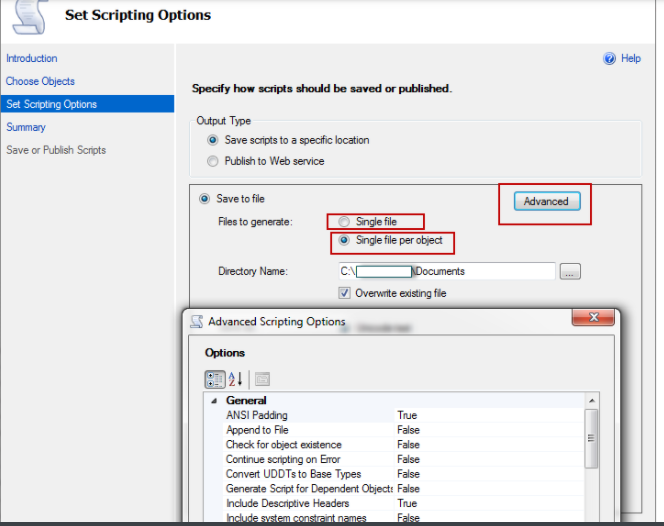
Ara yüzden veya komut satırından sorgular ile iki yol ile güncelleme işlemi yapabiliriz. Ara yüz ile yapabilmek için tablodan sağa tıklayıp colomn bölümünden bir tane daha sütun ekleyip veri tipi ile sonradan ekleme yapabiliriz. Ve allow null seçeneğini işaretlemek zorundayız yoksa hata verir. Komut ile yaparsak; new query den üzerine ekleme yapacağımız database in isminde çalışacağız. Kullanılacağımız komut alter komutudur nesneleri güncellememizi sağlar. verileri güncellemek için ise update komutu gerçekleştirilir.örnek: alter table deneme1 add sütun1 int

**Tetikleyici** ve **.bak** işlemine göre yedek alma örneği;

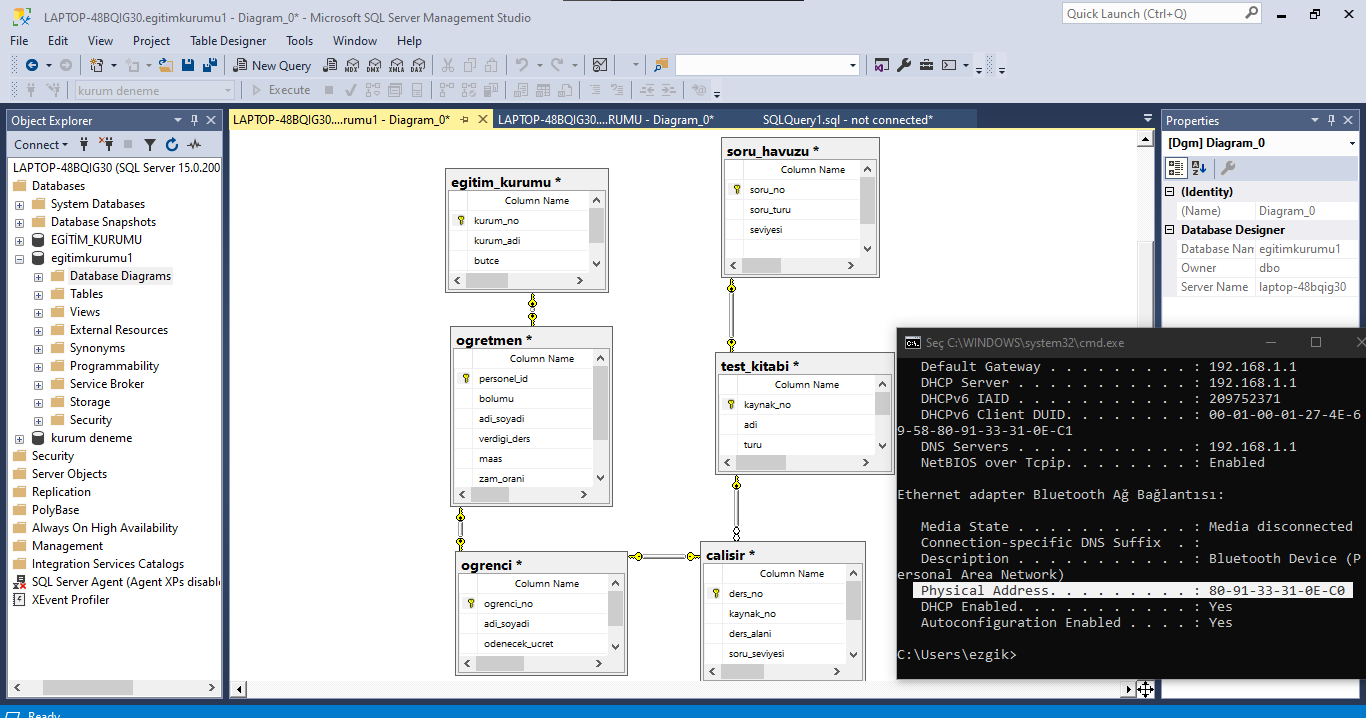
CREATE TRIGGER deneme

ON denemetablo AFTER INSERTAS BEGIN **COMMIT TRANSACTION** BACKUP DATABASE test TO DISK = N**'C:\deneme.bak'** WITH NOFORMAT, NOINIT, NAME = N'pcrm-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10 BEGIN TRANSACTIONENDGO

Tüm depolanmış prosedürleri, tetikleyicileri ve işlevleri tek bir sunucudan komut dosyası haline getirmek ve bunları yeni sunucuda oluşturmak istiyorsunuz. Bunun için;  



**6) Sizden bir veri tabanı oluşturmanız istenmektedir oluşturacağınız veri tabanının ilişki diagramının ekran görüntüsünü bu sorunun cevabı olarak ekleyiniz. (KONUSU: En az 6 tabloluk Özel Bir Eğitim Kurumunun kendi öğrencileri için hazırladığı test kitaplarını ve soru havuzlarını kontrol eden uygulama için veri tabanı.)**



**NOT: mac ve ya physical addres alma Başlat+R - cmd - ipconfig/all yazılarak en alttaki çıkana göre alınmıştır.**  **Soru 3 ve 6 nın ekran görüntüleri istenmekte.**