**ANKARA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**amblem, daire, logo, simge, sembol içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**BLM4522-Ağ Tabanlı Paralel Dağılım Sistemleri**

**PROJE AÇIKLAMASI : “Veritabanı Güvenliği ve Erişim Kontrolü” , “Veri Temizleme ve ETL Süreçleri Tasarımı” ve “Veritabanı Yedekleme ve Otomasyon Çalışması“ konularını uygulayarak gösterme**

**Ders Danışmanı : Enver BAĞCI**

**Elif BELLER - 19290957**

**Github hesabım : ElifBeller-eng**

**Proje Linkim : https://github.com/ElifBeller-eng/Ag-Tabanl-vize-sql-vs.**

**25.04.2025**

**İÇİNDEKİLER**

1. **Veritabanı Güvenliği ve Erişim Kontrolü**

**Amaç ...................................................................................3**

**Çözüm - Örnek .....................................................................4**

**Ek notlar -SQL Server Audit Özelliği ......................................5**

**SQL Server Audit Özelliği ......................................................6**

1. **Veritabanı Güvenliği ve Erişim Kontrolü**

**Amaç - Teorik Bilgi - Senaryo ...............................................7**

**Uygulama Adımları ve SQL Kodları .......................................8**

**Temiz Verileri Yeni Tabloya Aktarma (LOAD Kısmı) - Sonuç ..9**

**Ek Notlar - Örnek ................................................................10**

1. **Veritabanı Yedekleme ve Otomasyon Çalışması**

**Amaç - Yedekleme Türleri - Senaryo ...................................11**

### ****Yedekleme SQL Komutu - SQL Server Agent ile Otomasyon - Kazanımlar ve Sonuç - Ek Bilgiler ........................................12****

**Geri Yükleme (Restore) Testi - Otomasyon ......................13**

**Zamanlama (Task Scheduler) ...................................14-15-16**

**Test ...................................................................................17**

**Sonuç ................................................................................18**

💻🕒

**1.Veritabanı Güvenliği ve Erişim Kontrolü**

## ****Amaç :****

Verilerin yetkisiz erişime karşı korunması, kullanıcı rollerinin sınırlandırılması ve olası SQL Injection saldırılarına karşı önlem alınması.

Veritabanı güvenliği, kullanıcı verilerinin gizliliğini ve bütünlüğünü korumak için alınan önlemleri kapsar. En yaygın tehditlerden biri SQL injection saldırılarıdır. Ayrıca kullanıcıların yetkilerini sınırlandırmak ve erişim denetimi sağlamak için rol tabanlı izinler tanımlanmalıdır.

**Temel başlıklar :**

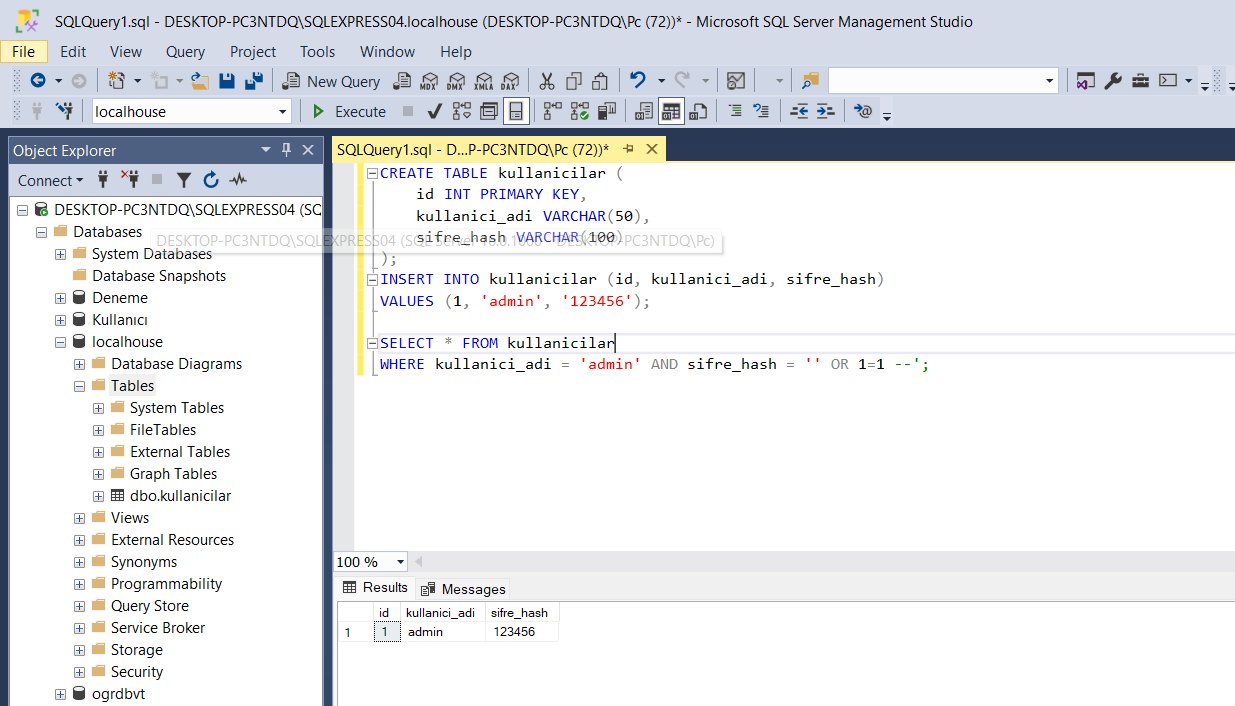
- SQL Injection ve korunma yolları

- Yetkilendirme (GRANT, REVOKE)

- Kullanıcıların rollerle sınırlandırılması

- Hassas verilerin şifrelenmesi (örn. şifre alanları)

(SQL Injection Saldırı Örneği) Tüm kullanıcılar döner, güvenlik açığı oluşur. :

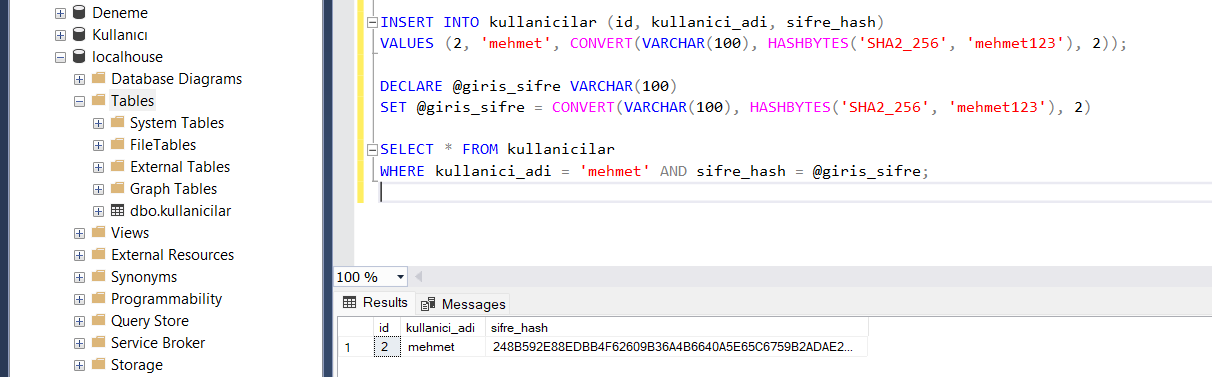


**Çözüm :**

Giriş sistemi, dışarıdan gelen değerleri filtrelemeli. Kodun bu kısmı genelde yazılım tarafından parametreli gönderilir.

Şifreleri güvenli saklama (T-SQL ile hashleme)

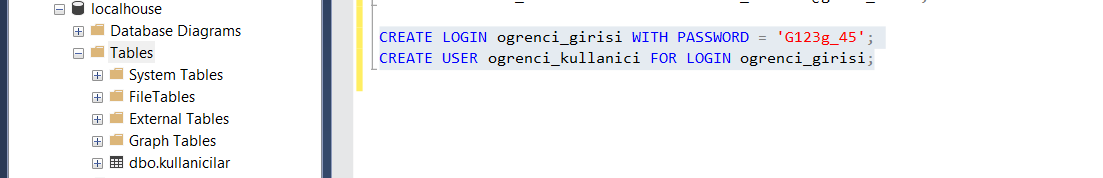
Yeni kullanıcı eklerken HASHBYTES ‘le; şifreyi SHA-256 ‘le saklayalım



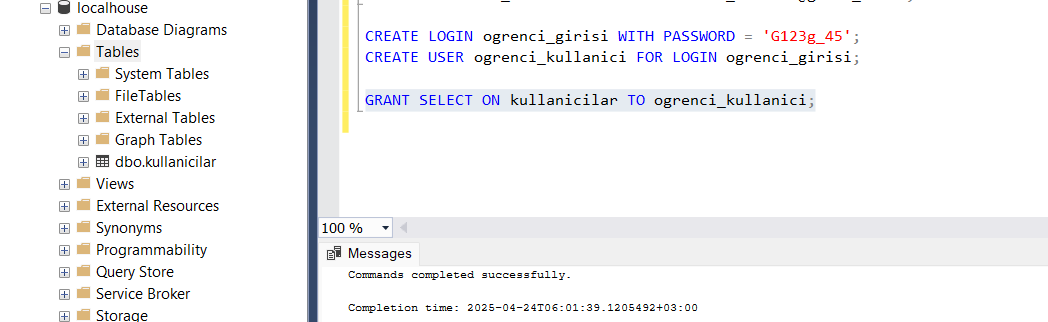
(Hash fonksiyonuyla şifre saklama ve sonrasında da giriş kontrolü)

Bu projede veritabanı güvenliğinin temel prensipleri uygulanarak, hem kullanıcı erişimleri sınırlandırılmış hem de şifrelerin güvenli biçimde saklanması sağlanmıştır. Ayrıca SQL injection gibi saldırı türlerine karşı örnekli çözümler gösterilmiştir.

**Rol Tabanlı Erişim Örneği :**



Sadece veri görme izni verelim :



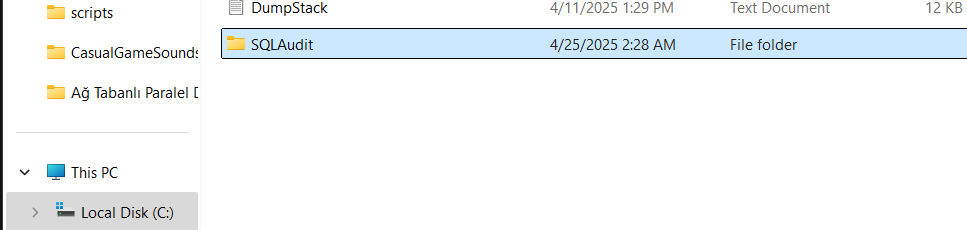
**Ek Notlar :**

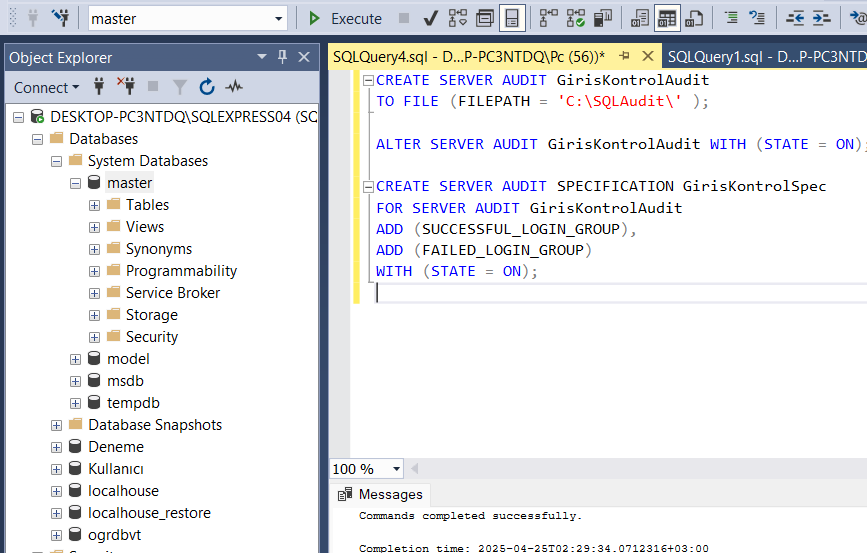
Gerçek projelerde veritabanına erişim öncesi input validation ve parametreli sorgular (prepared statements) kullanılmalıdır.

Şifreleme işlemi tek yönlü hashleme ile yapılmalı, çözülemeyen algoritmalar tercih edilmelidir.

**SQL Server Audit Özelliği**

Eğer güvenlik kısmında kimin ne zaman ne işlem yaptığını izlemek istiyorsak, bunun için SQL Server Audit kullanırız.





Bu kodla :

- Audit nesnesi oluştur.

- Başlat

- Audit Specification (örnek: login işlemlerini izle)

Bu komutlarla başarılı ve başarısız login denemeleri izlenir. Loglar C:\SQLAudit klasörüne kaydedilir. Varsa D sürücüsü de kullanılabilir.

Audit dosyalarını sadece yetkili kullanıcılar görebilir.

**2.Veri Temizleme ve ETL Süreçleri Tasarımı**

**Amaç :**  
Dağınık ve eksik veri kaynaklarını temizleyerek anlamlı hale getirmek, dönüşümden geçirip hedef sisteme aktarmak.

**Teorik Bilgi :**  
ETL (Extract - Transform - Load) süreci, veri ambarları ve analiz sistemleri için temel yapı taşlarından biridir:

**- Extract (Çıkarma) :** Verilerin kaynak sistemlerden alınması

**- Transform (Dönüştürme):** Verilerin temizlenmesi, biçimlendirilmesi, dönüştürülmesi

**- Load (Yükleme):** Dönüştürülen verilerin hedef sisteme yüklenmesi

Veri Temizleme işlemleri genellikle aşağıdaki işlemleri içerir:

- Boş alanların tespiti ve temizlenmesi

- Hatalı formatların düzeltilmesi

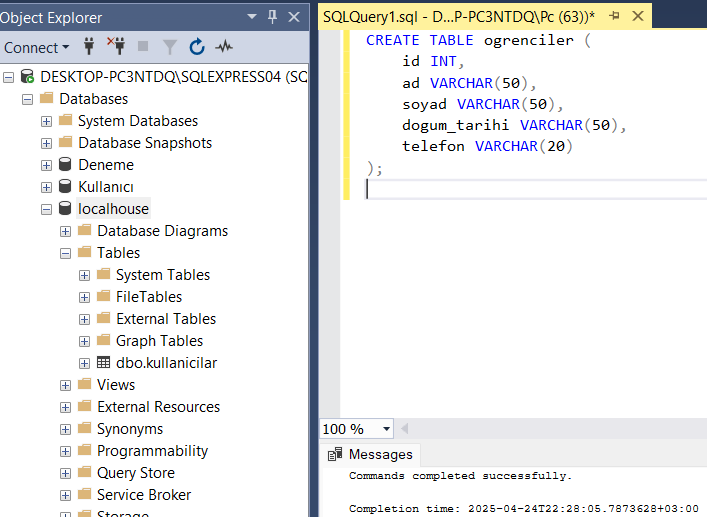
- Tekrarlayan kayıtların ayıklanması

- Uyuşmayan tarih/sayı/veri tiplerinin düzeltilmesi

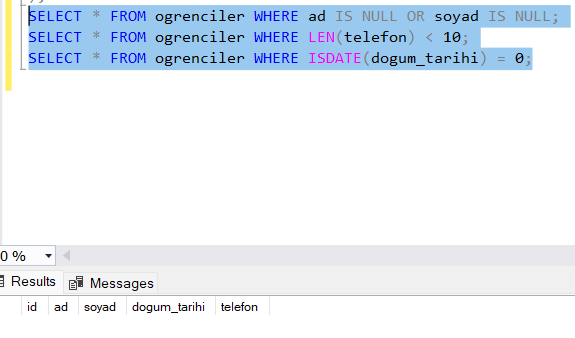
**Senaryo :**  
Bir okul sisteminden alınan öğrenci bilgilerinin yer aldığı tablo hatalı ve eksik veriler içermektedir.  
Bu veriler temizlenip düzgün formata getirilerek raporlama sistemine aktarılacaktır.

**Uygulama Adımları ve SQL Kodları:**

1. **Kaynak Tablo Örneği (ogrenciler)**



1. **Hatalı ve Eksik Verilerin Tespiti**

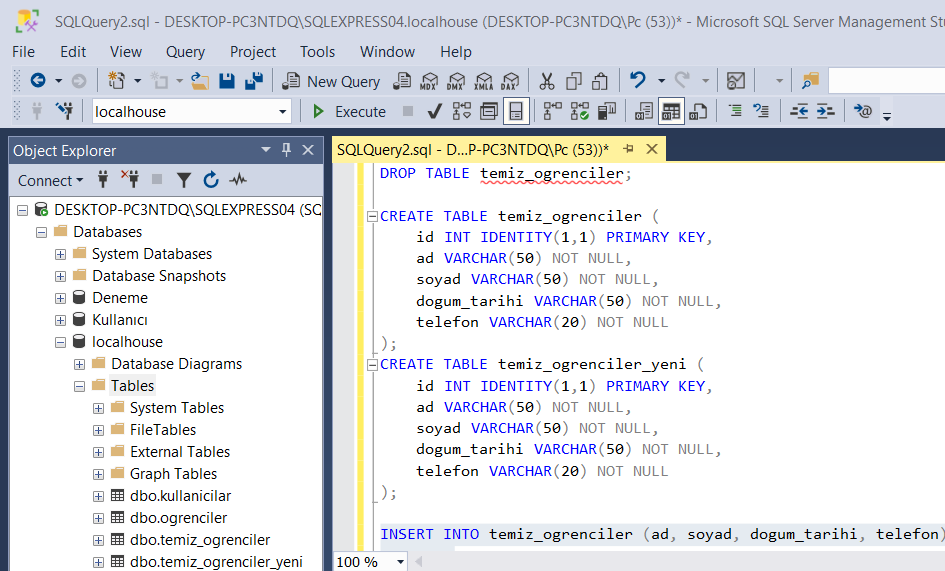


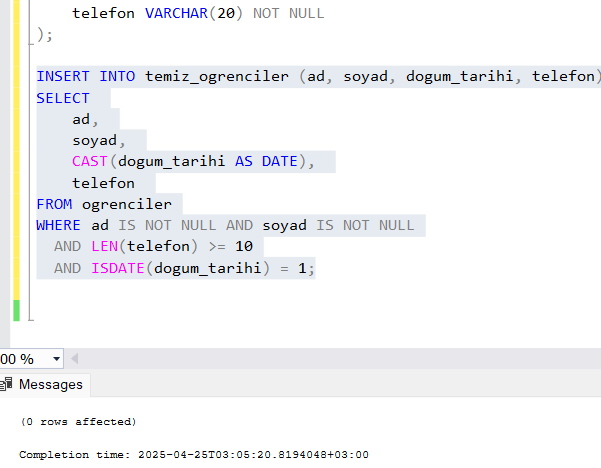
- Boş ad veya soyad

- Geçersiz telefon uzunluğu

- Geçersiz tarih formatı kontrolü (örnek)

1. **Temiz Verileri Yeni Tabloya Aktarma (LOAD Kısmı)**





Geçerli verileri taşıma

**Sonuç :**  
Bu projede veri temizliği gerçekleştirilmiş, hatalı ve eksik kayıtlar filtrelenmiş, düzgün veriler hedef tabloya yüklenmiştir. ETL süreci sayesinde veri analizi için daha anlamlı ve tutarlı bir veri kümesi elde edilmiştir.

**Ek Notlar :**

- Gerçek projelerde veri temizliği SSIS (SQL Server Integration Services) gibi araçlarla otomatikleştirilebilir.

- Temiz veri, doğru raporlamanın temelidir.

- Bu işlemler büyük veri ortamlarında hayati öneme sahiptir.

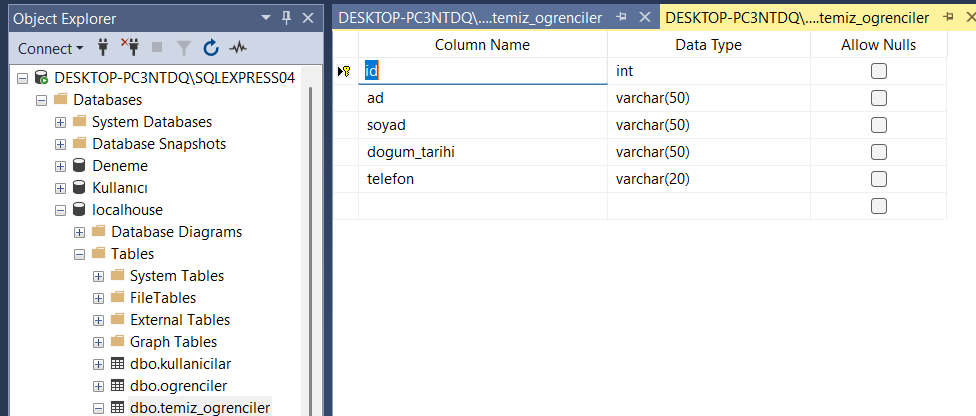
**Örnek :**

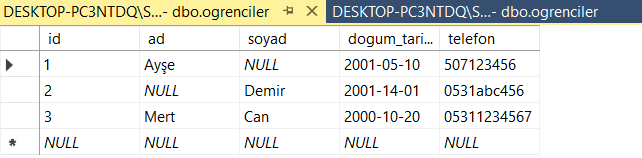
**ETL Öncesi - Önceki Veri (ogrenciler)**

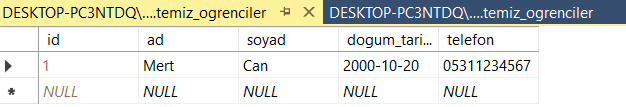
| **id** | **ad** | **soyad** | **dogum\_tarihi** | **telefon** |
| --- | --- | --- | --- | --- |
| 1 | Ayşe | NULL | 2001-05-10 | 507123456 |
| 2 | NULL | Demir | 2001-14-01 | 0531abc456 |
| 3 | Mert | Can | 2000-10-20 | 05311234567 |

**ETL Sonrası - Temizlenmiş Veri (temiz\_ogrenciler)**

| **id** | **ad** | **soyad** | **dogum\_tarihi** | **telefon** |
| --- | --- | --- | --- | --- |
| 3 | Mert | Can | 2000-10-20 | 05311234567 |







**3.Veritabanı Yedekleme ve Otomasyon Çalışması**

**Amaç :**

- Verilerin kaybolma riskine karşı korunması

- Yedekleme işlemlerinin otomatik hale getirilmesi

- Sistem yöneticisinin iş yükünün azaltılması

Özetle : Güvenli ve sürdürülebilir veri yönetimi

### ****Yedekleme Türleri :****

**- Tam Yedek (Full) :** Veritabanının tümü

**- Fark Yedeği :** Son tam yedekten sonra değişen kısımlar

**- Transaction Log :** Her değişiklik adım adım izlenir

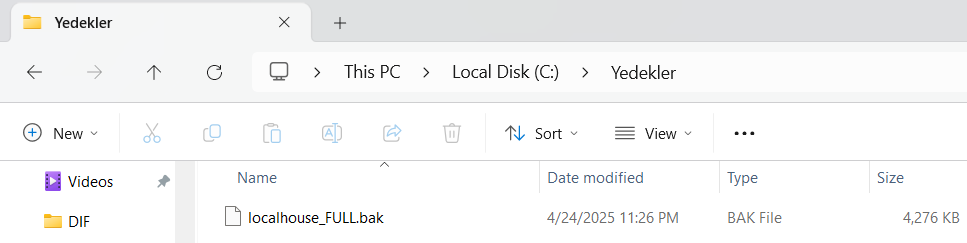
Bu yedekleme türleri birlikte kullanılabilir.

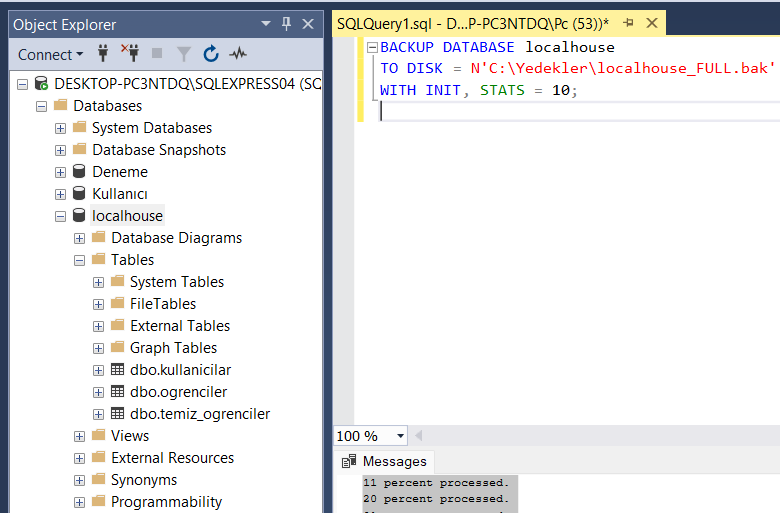
### ****Senaryo :****

“Her gece saat 02:00'de okul veritabanı otomatik olarak yedeklenmelidir.”  
Sistem yöneticisi manuel işlem yerine bir otomasyon kurar.

**Yedekleme SQL Komutu :**

SQL ile doğrudan tam yedek alınır.





### ****SQL Server Agent ile Otomasyon :****

1. SQL Server Agent → New Job
2. Job adı : **Gece\_Yedekleme**
3. Step : BACKUP DATABASE komutu
4. Zamanlama : **02:00** günlük
5. Job aktif hale gelir → Otomatik yedekleme başlar

### ****Kazanımlar ve Sonuç :****

- Veri kaybı riski azalır

- Süreklilik ve güvenlik sağlanır

- Zaman kazancı elde edilir  
Otomasyon, sistem yönetiminde verimliliği artırır.

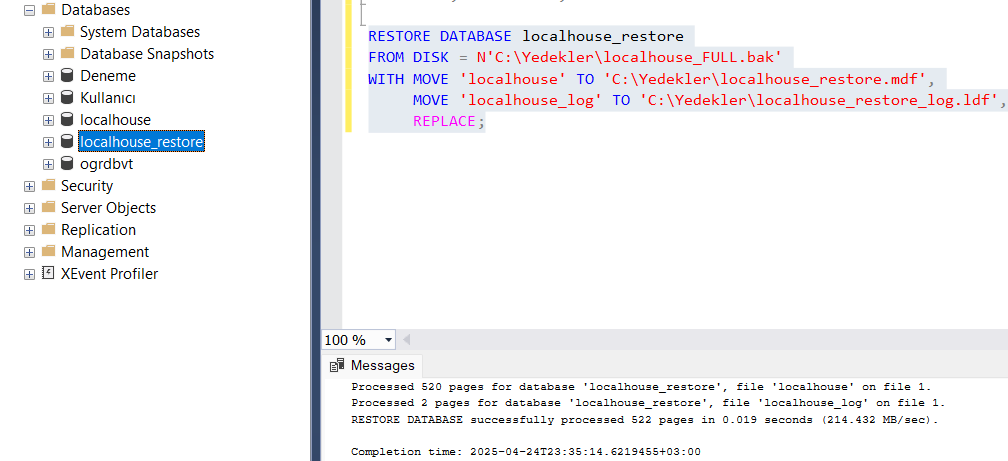
### ****Ek Bilgiler :****

- Yedeklerin başka diskte/uzakta tutulması önerilir

- RESTORE DATABASE işlemi test edilmelidir

- PowerShell veya Batch Script ile gelişmiş otomasyon yapılabilir

-- localhouse\_FULL.bak dosyası varsa, **Geri yükleme (restore) testi** yapabiliriz. :



Veritabanını geri yüklemek için (başka bir isimle istersen, mesela “localhouse\_restore”)

**Not :** Veritabanı açıkken RESTORE işlemi çalışmaz, ya önce bağlantıları kapatmalıyız ya da başka bir adla yüklemeliyiz.

**Otomasyon**

Belirli bir zaman aralığında sistemin kendi kendine bir işi yapmasıdır.

Ve burada o iş: **Veritabanını yedekle.**

**otomasyon kısmı = zamanlama + otomatik çalıştırma** diyebiliriz.

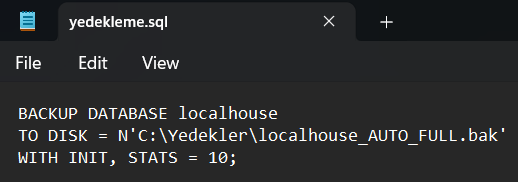
### ****Otomasyonun İçeriği :****

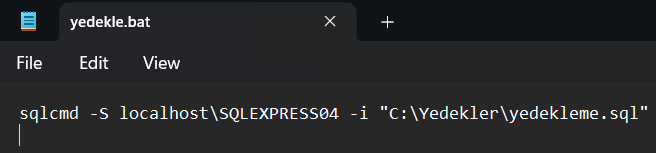
SQL Server Agent'ın kullanılmadığı Express sürümüne göre alternatif çözüm : (Task Scheduler + .bat)

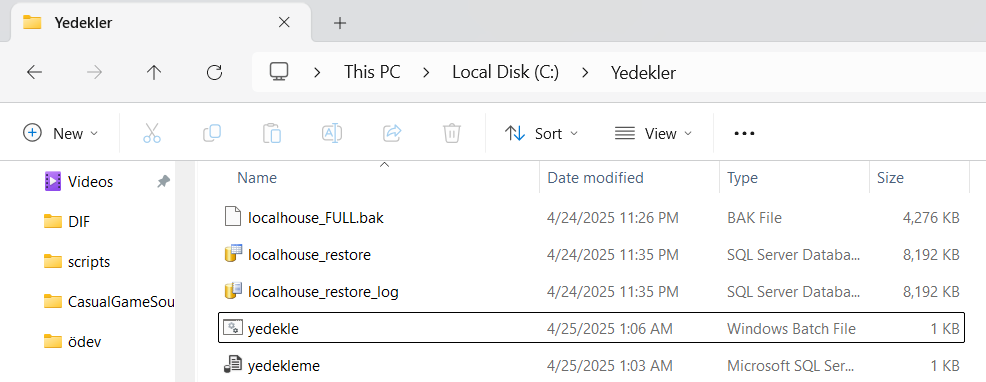
1. **SQL komutunu** bir .sql dosyasında saklamak (.sql dosyasının hazırlanması)  
   Hedef : yedekleme komutunu dışarıdan çağırabilmek

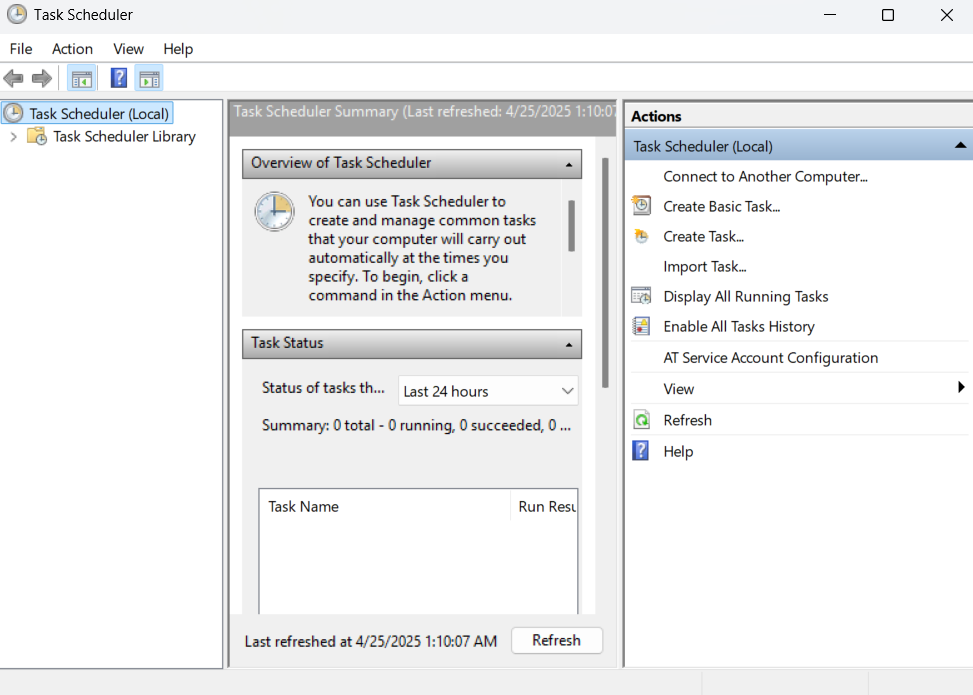
**.bat dosyası** ile bu komutu MSSQL'e göndermek (.bat ile çalıştırılması)  
Hedef : elle SSMS açmadan otomatik çalıştırmak

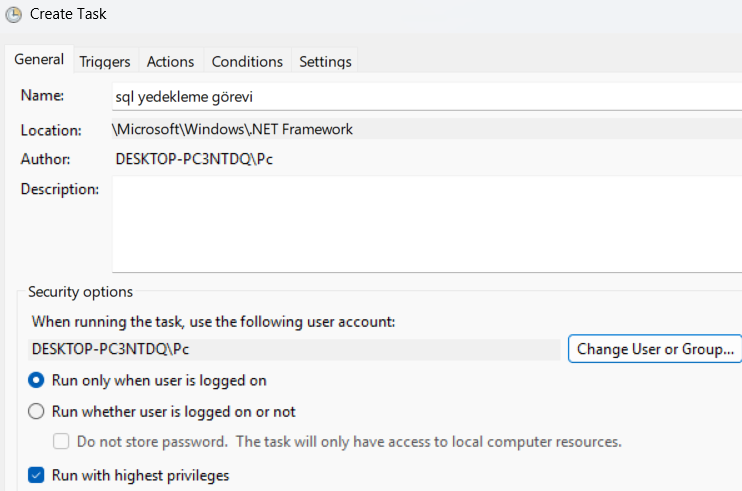
**Zamanlama (Task Scheduler) :** Görev Zamanlayıcı ile **zamanlama (schedule)** yapılmasıdır.  
Hedef : .bat dosyasını **her gün saat 02:00'de** otomatik çalıştırmak

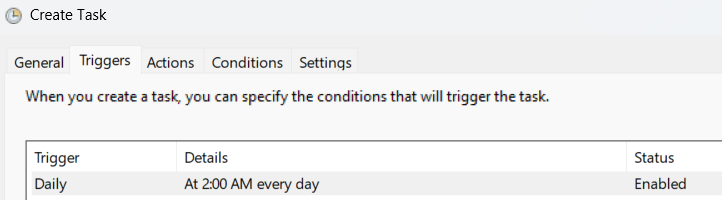


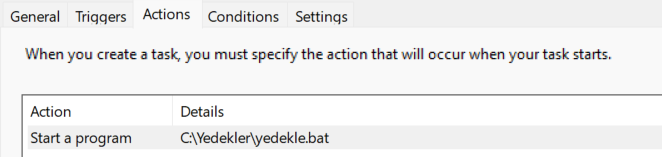


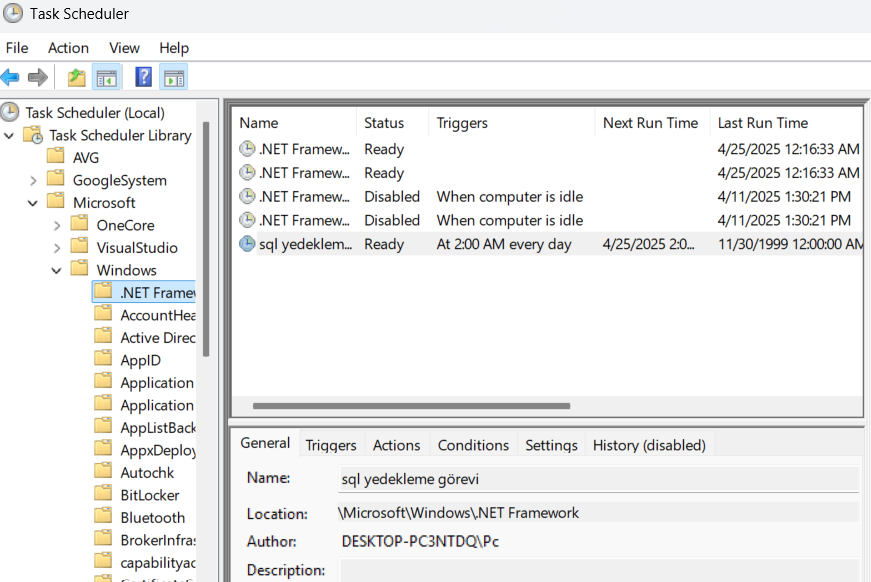






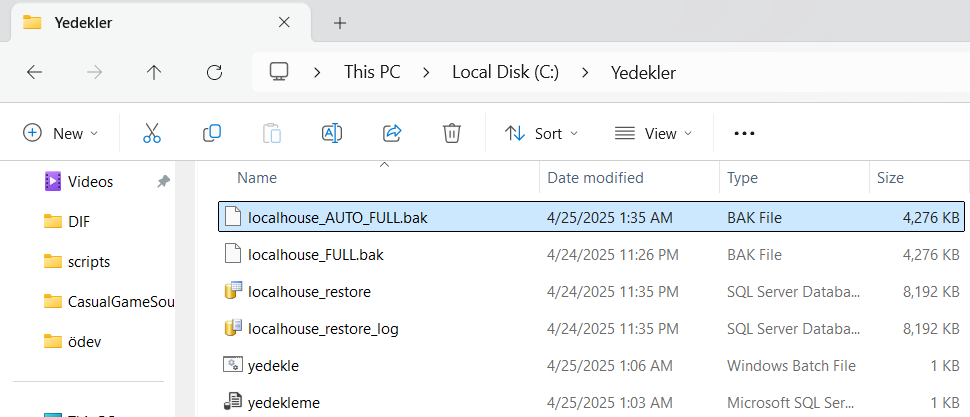


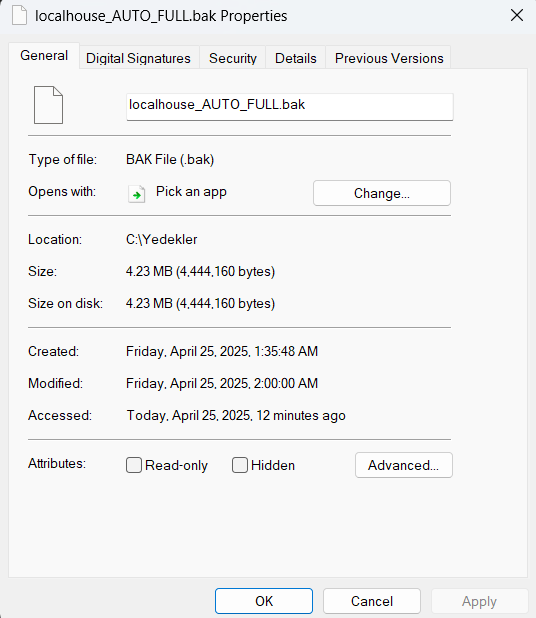




## Test Et :

## Göreve sağ tıklayıp Run (Çalıştır) diyoruz :





**Sonuç :**

Veritabanı yedekleme sürecini otomatikleştirmek amacıyla Windows Görev Zamanlayıcı (Task Scheduler) kullanılmıştır. SQL Server Agent özelliği Express sürümünde bulunmadığı için .sql komutu bir .bat dosyası aracılığıyla tetiklenmiş ve günlük olarak yedek alınması sağlanmıştır.