AĞ TABANLI PARALEL DAĞITIM SİSTEMLERİ VİZE PROJELERİ

AHMET EREN ÖZCAN

21290207

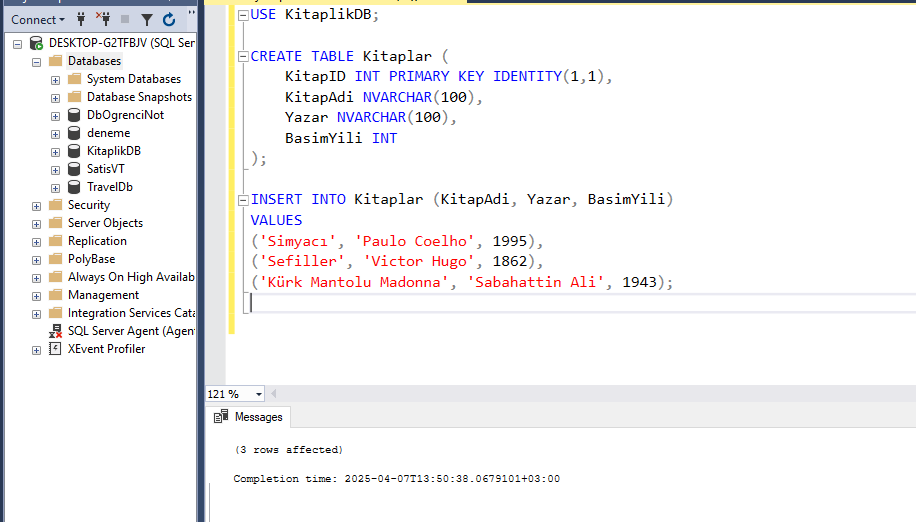
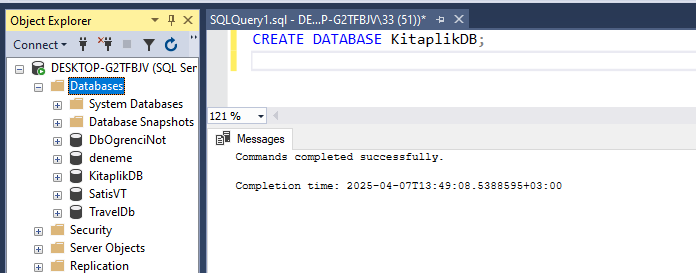
GİTHUB:ahmeterenozcan

**1-) SQL Server Veritabanı Yedekleme ve Otomasyon Projesi**

**Proje Adı:** KitaplikDB Günlük Yedekleme

**Adım 1: Veritabanı Oluşturma**

**Yapılan İşlem:**  
SSMS üzerinde KitaplikDB adlı bir veritabanı oluşturuldu. Daha sonra örnek bir Kitaplar tablosu eklendi ve örnek veriler girildi.

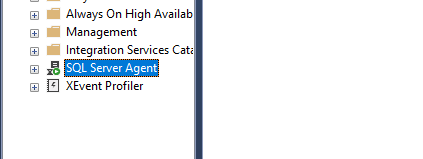


**Adım 2: Yedekleme Klasörü Oluşturma**

**Yapılan İşlem:**  
D:\SQL\_Backups\ klasörü oluşturuldu. SQL Server servis hesabının bu klasöre erişim izni verildi.

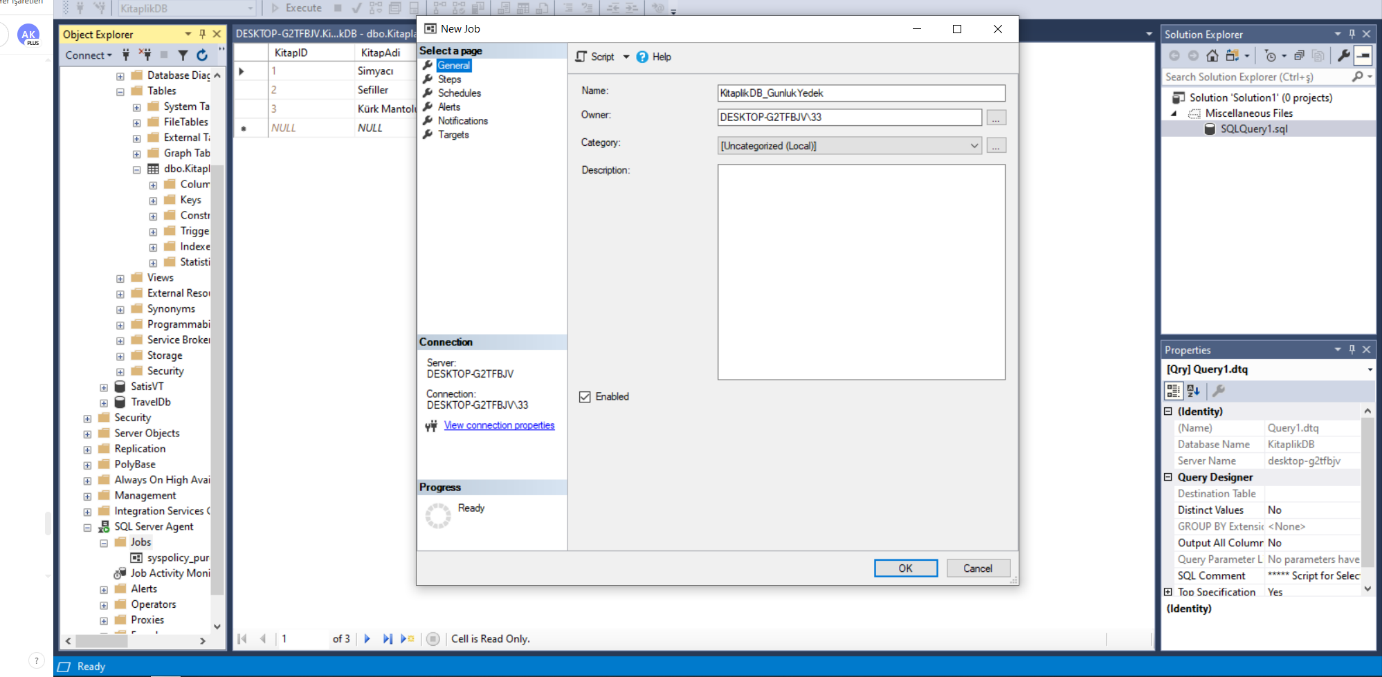
**Adım 3: SQL Server Agent’ı Başlatma**

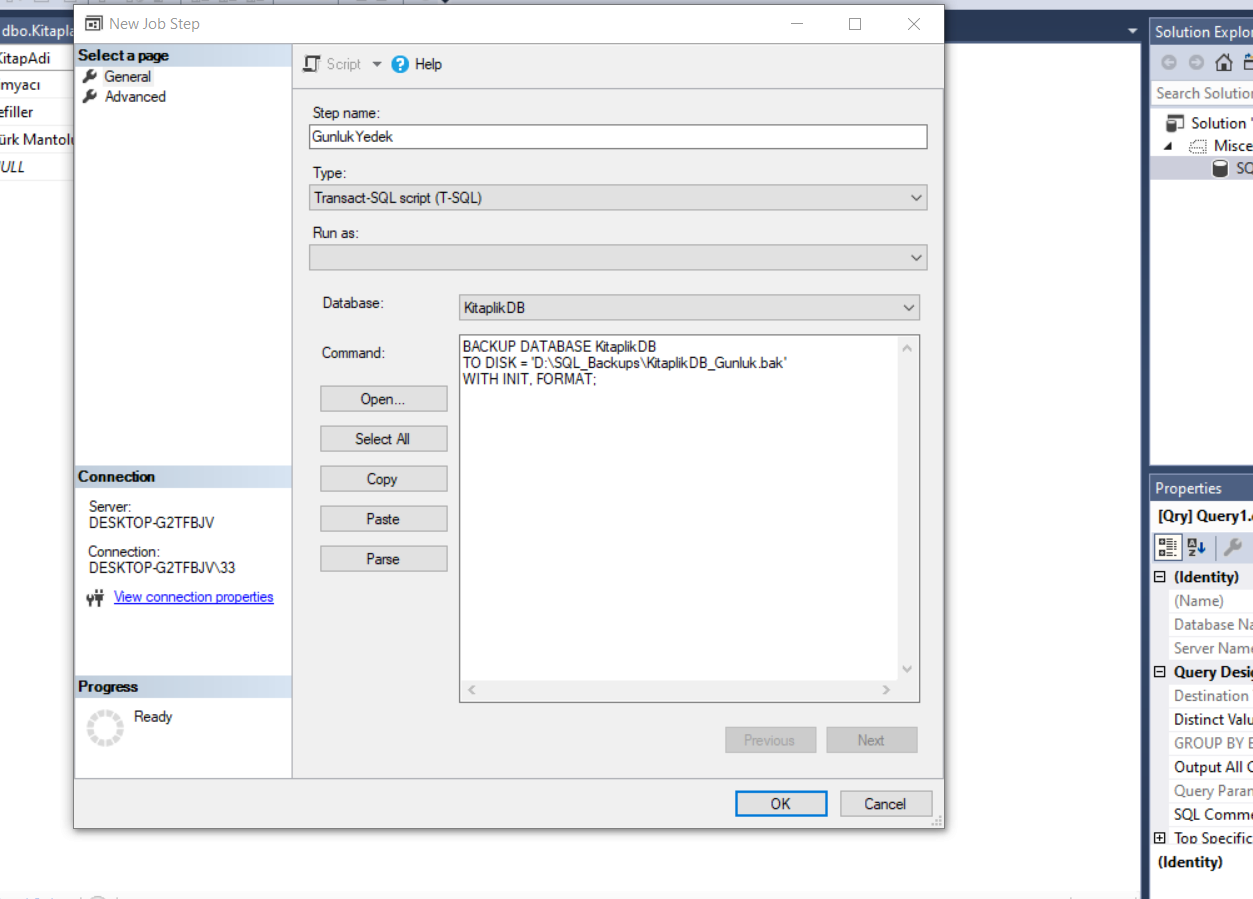
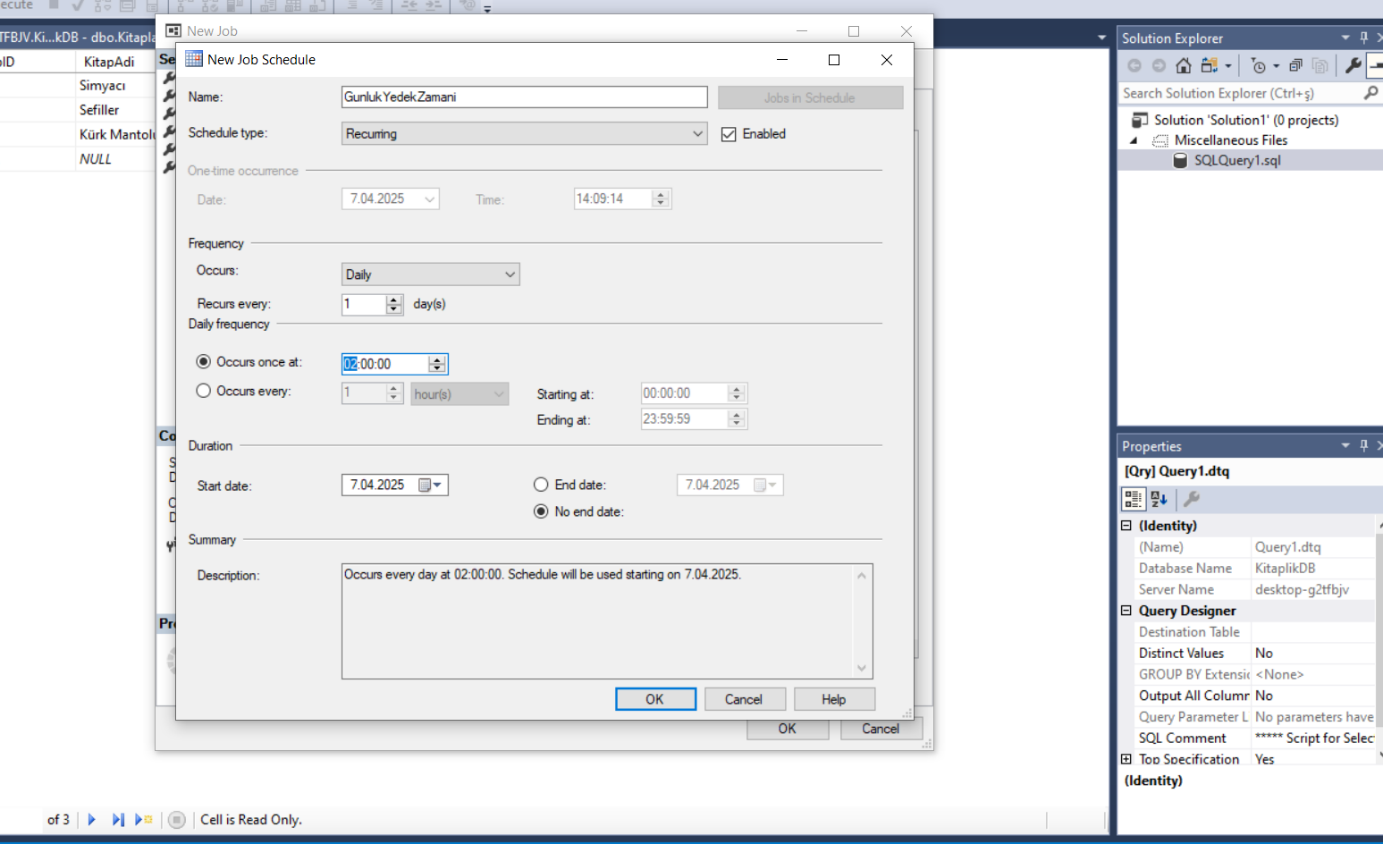
**Yapılan İşlem:**  
SQL Server Agent servisi başlatıldı. Bu sayede zamanlanmış görevler (Job'lar) çalıştırılabilir hale geldi.



**Adım 4: SQL Server Agent Job Oluşturma**

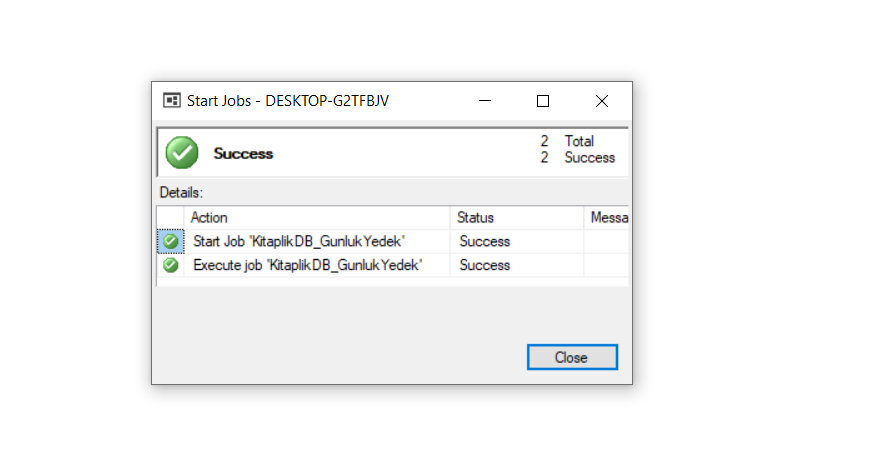
**Yapılan İşlem:**  
Bir Job oluşturularak, KitaplikDB veritabanı her gün otomatik olarak yedeklenecek şekilde ayarlandı.

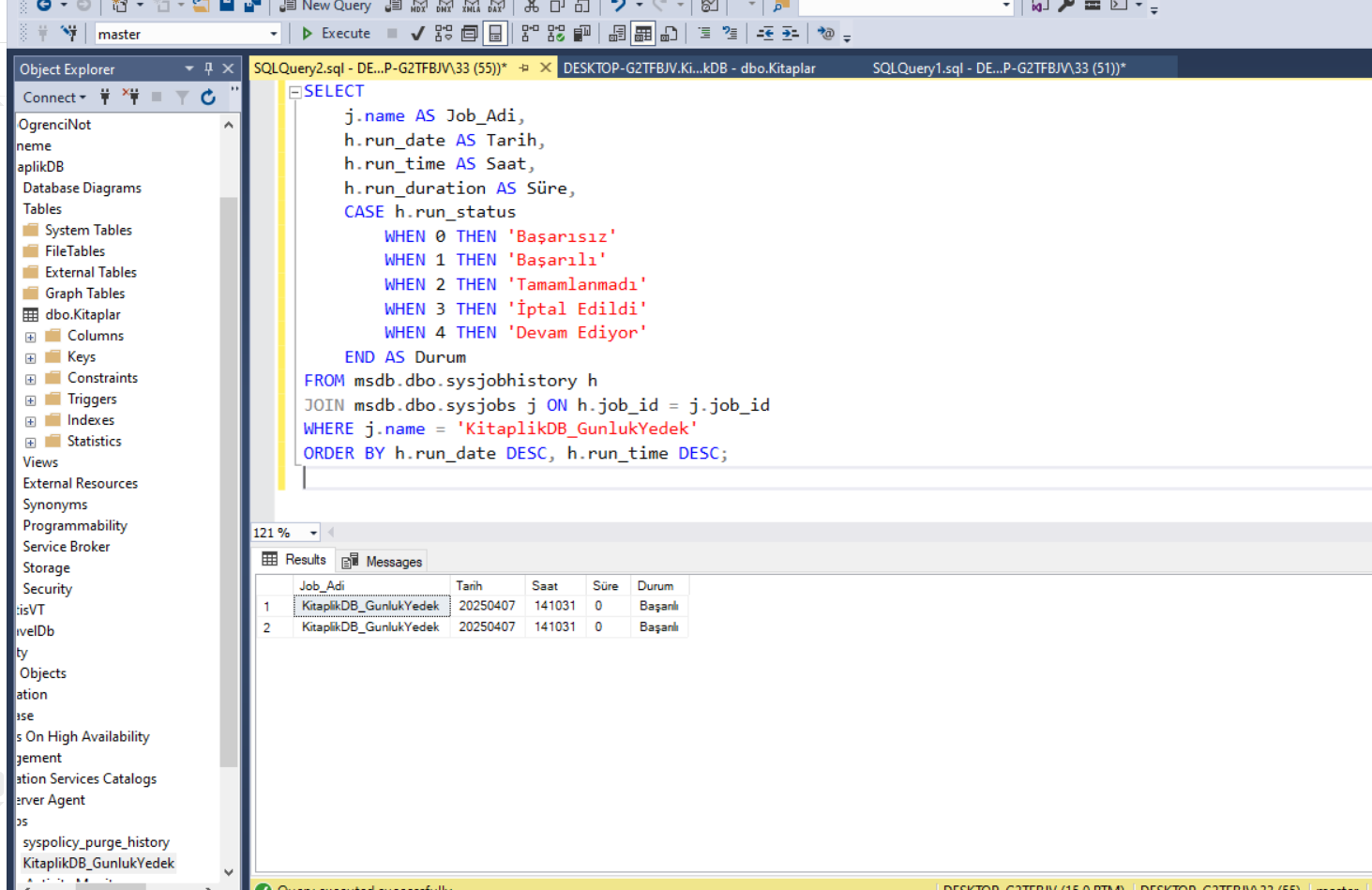




**Adım 6: Yedekleme Raporu Oluşturma**

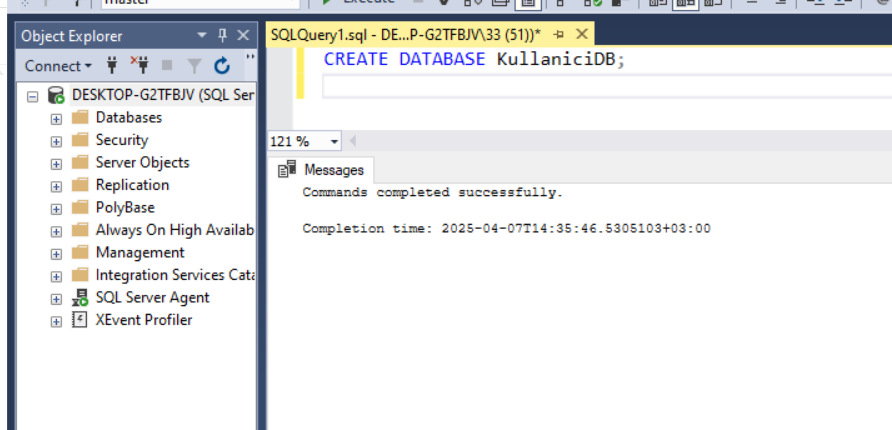
**Yapılan İşlem:**  
Job geçmişi sorgulandı ve yedeklemenin başarılı olup olmadığı izlendi.

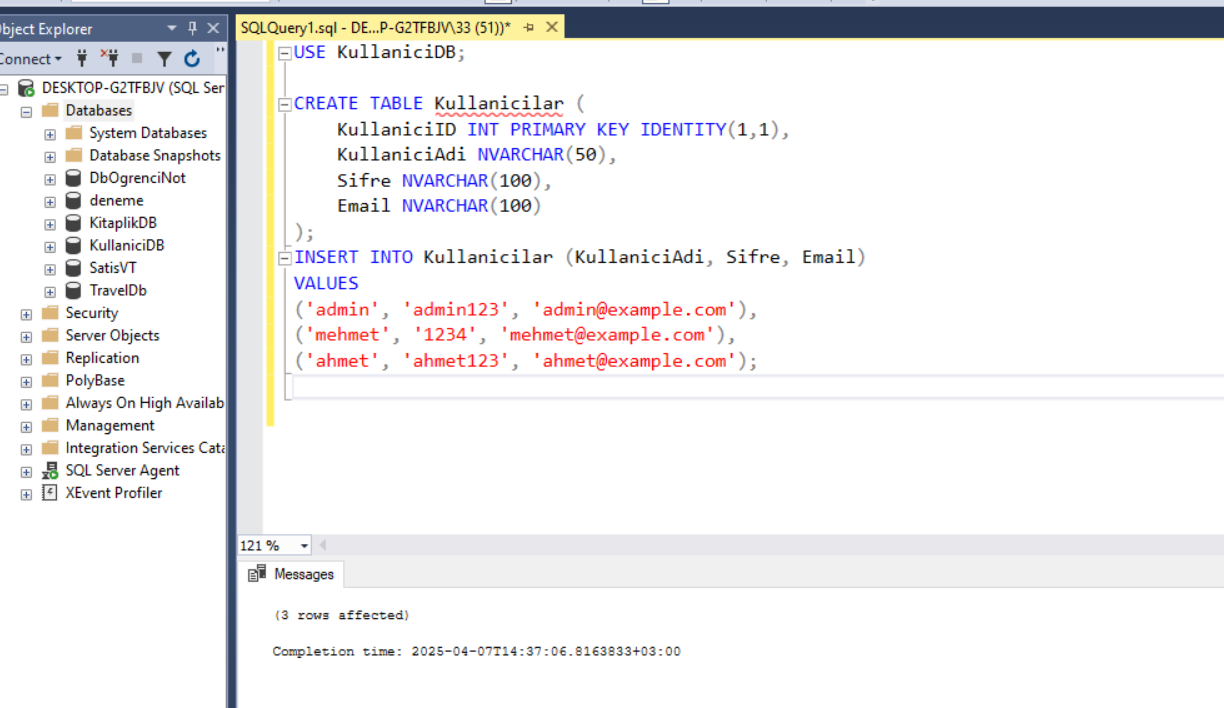




2-)Veritabanı Güvenliği ve Erişim Kontrolü

**Adım 1 – Veritabanı ve Tablo Oluşturma**

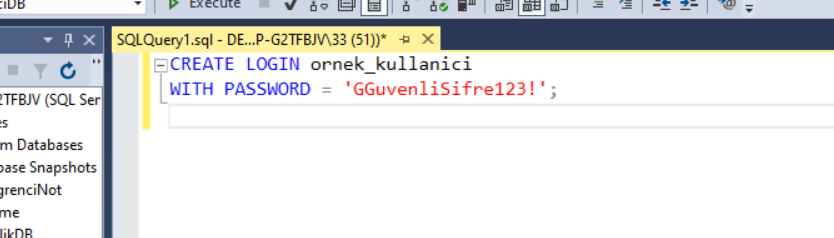
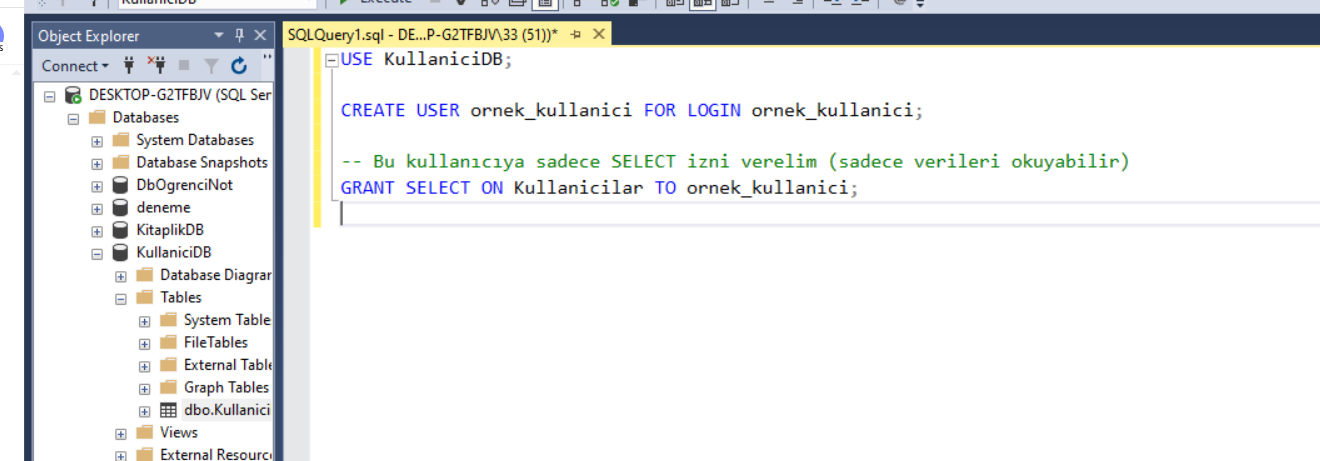
****KullaniciDB isimli bir veritabanı ve içerisinde Kullanicilar adlı tablo oluşturulmuştur.  
Tabloya örnek kullanıcı verileri girilmiştir.

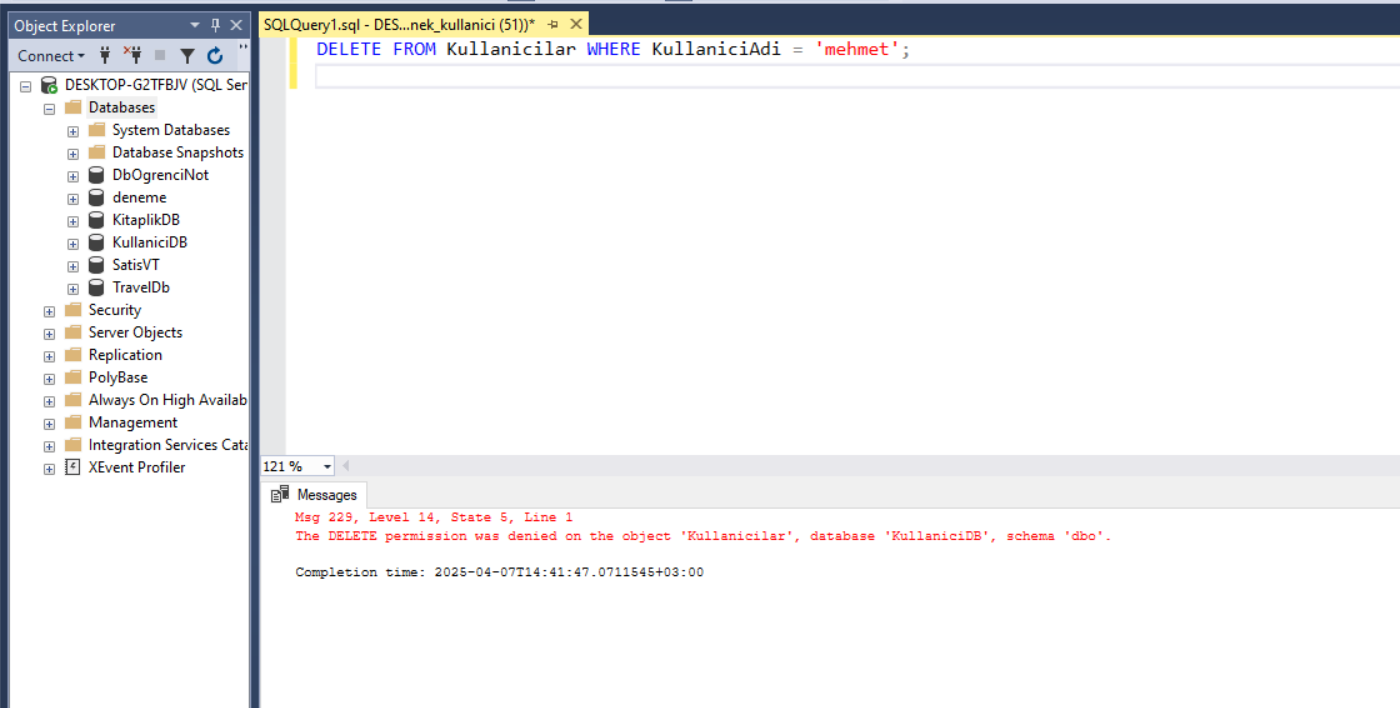


**Adım 2 – Kullanıcı Oluşturma ve Yetki Verme**

**Açıklama:**

ornek\_kullanici adlı bir SQL Server Login oluşturuldu ve sadece SELECT izni verildi.



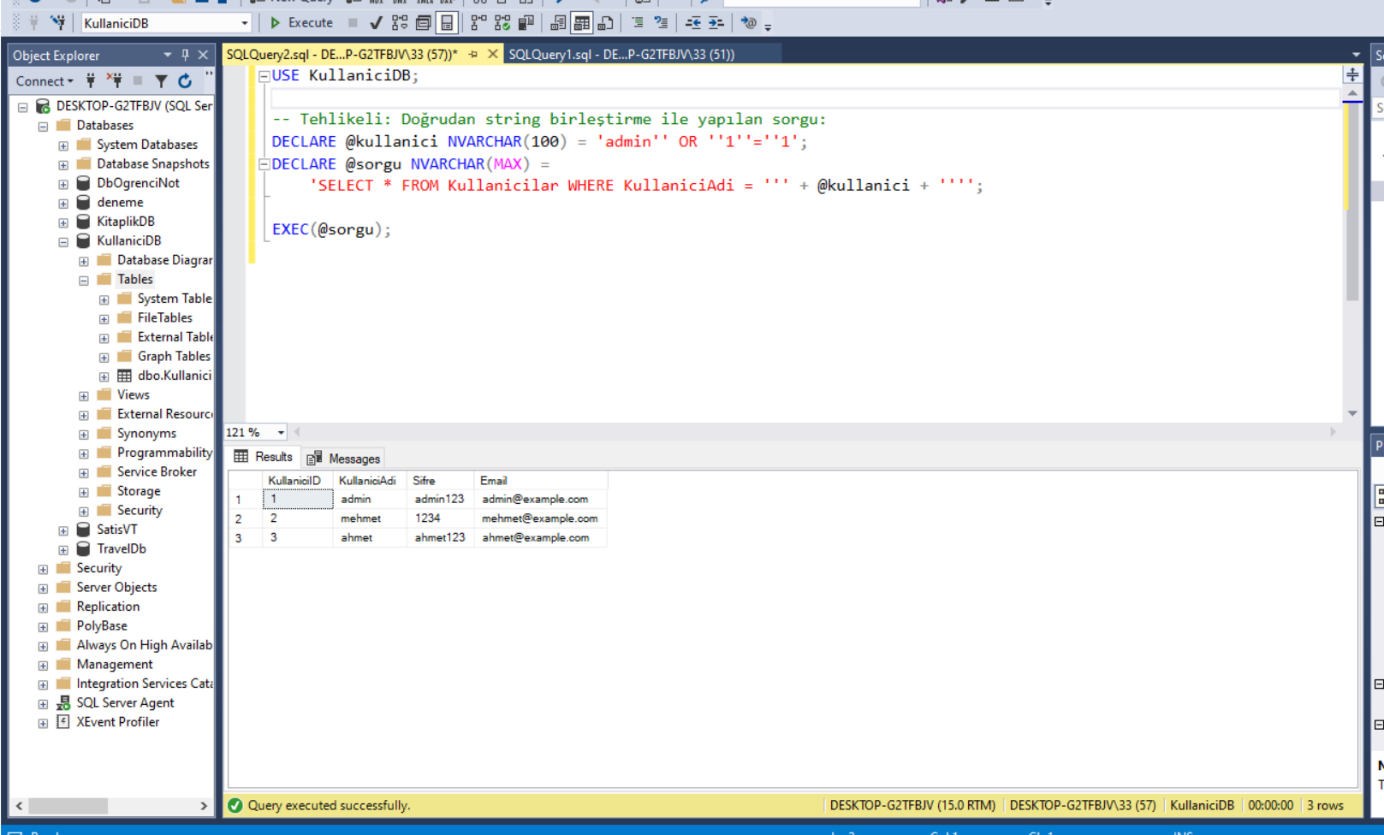
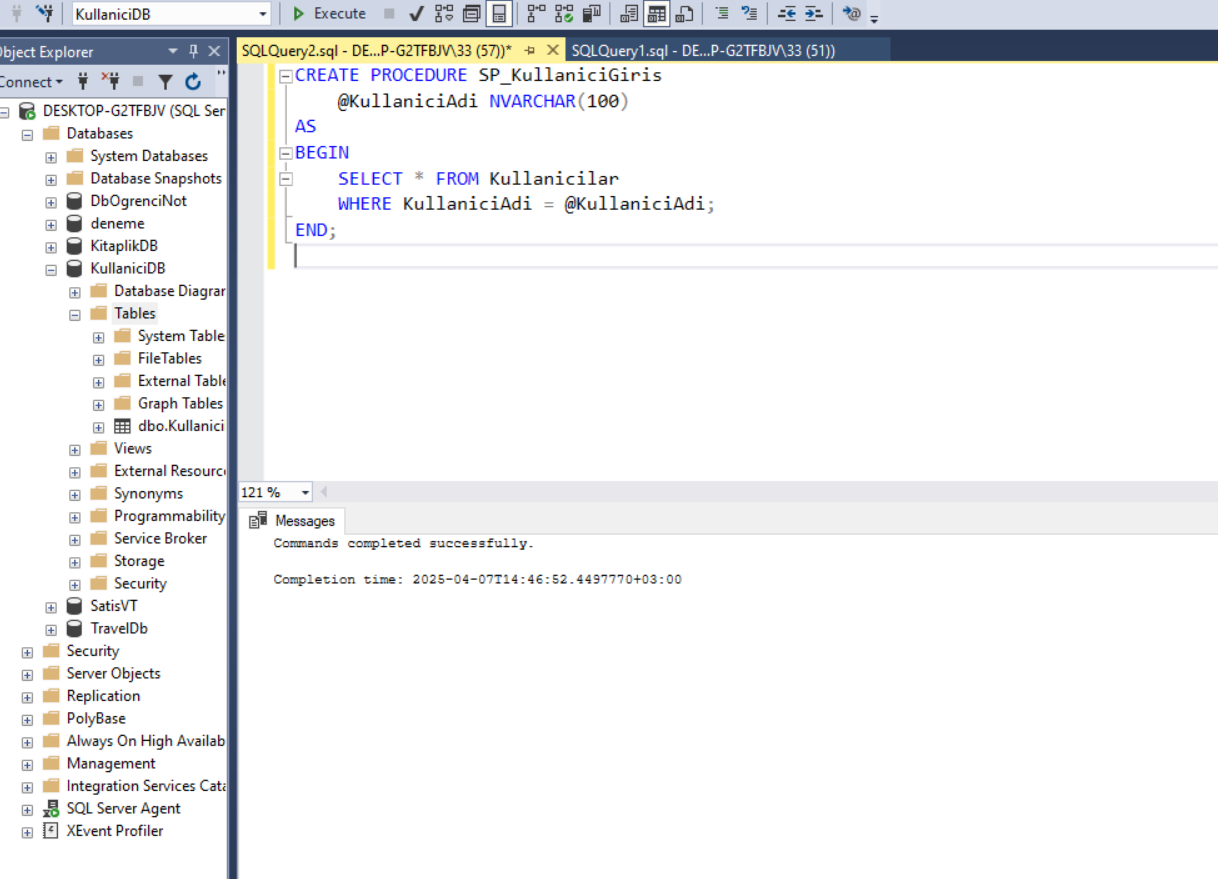


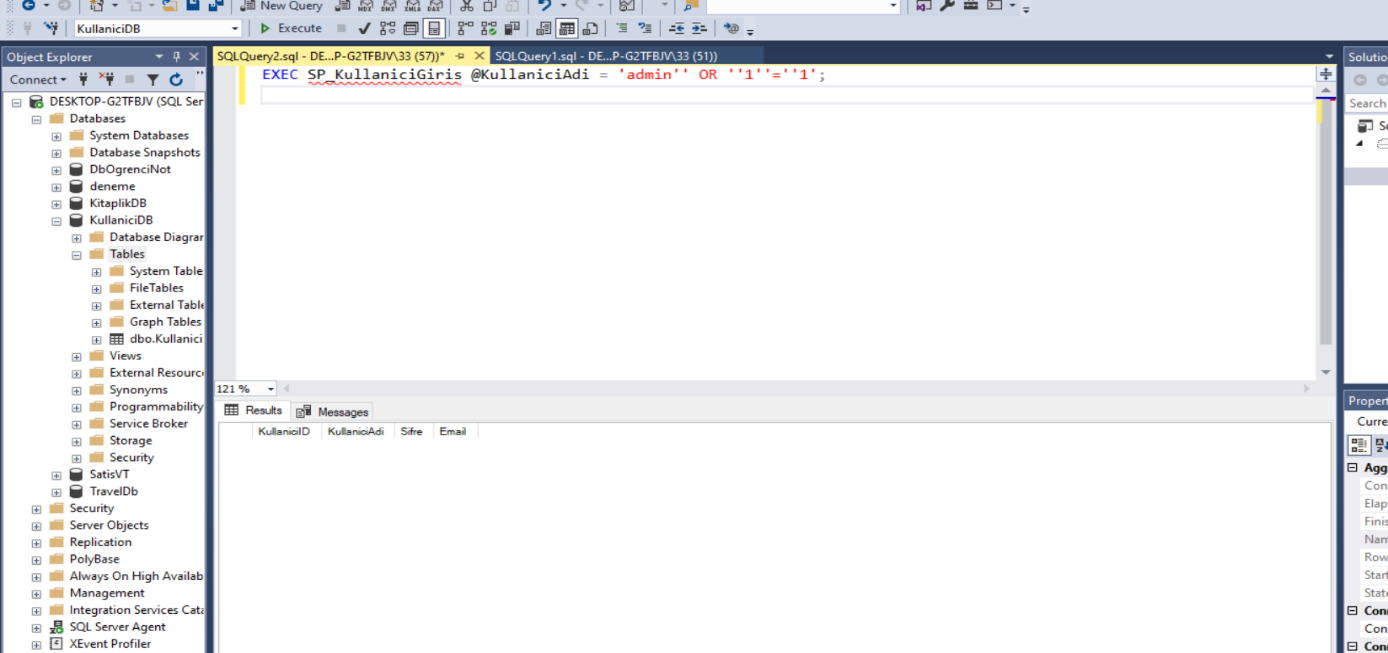
Burada ise yetkisiz sorgulama yapılıyor

**Adım 3 – SQL Injection Testi ve Korunma**

**Açıklama:**

SQL Injection saldırısı simüle edilerek, zararlı girişler test edilmiştir.  
Ardından güvenli sorgulama için Stored Procedure kullanılmıştır.



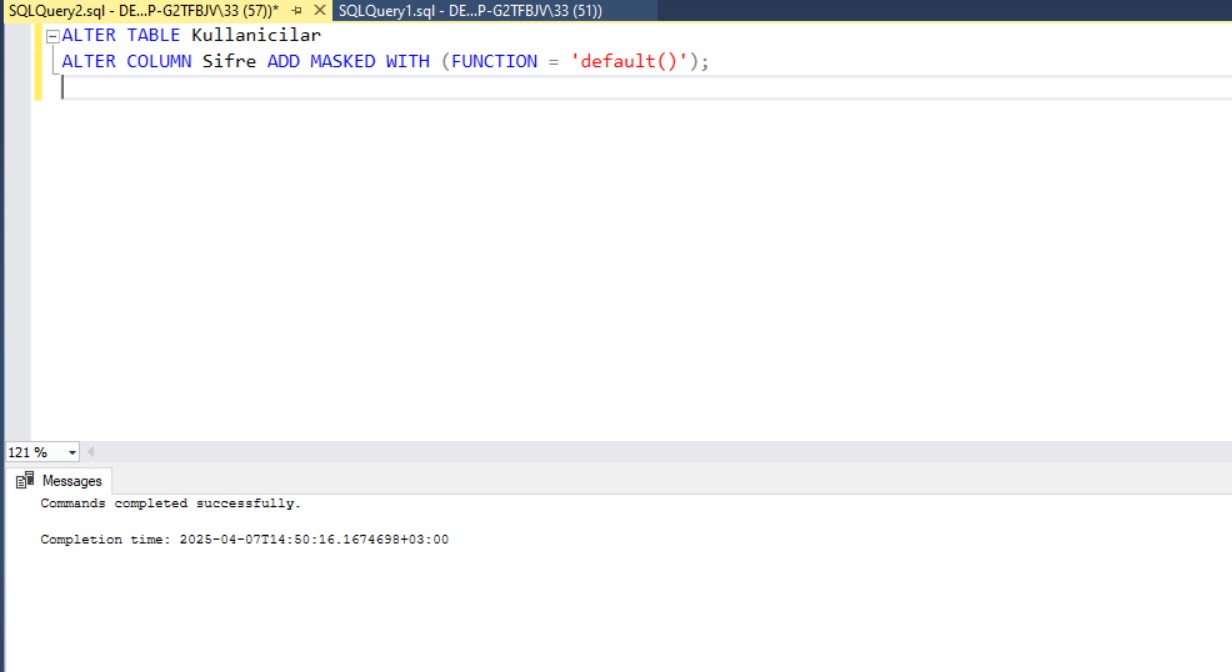


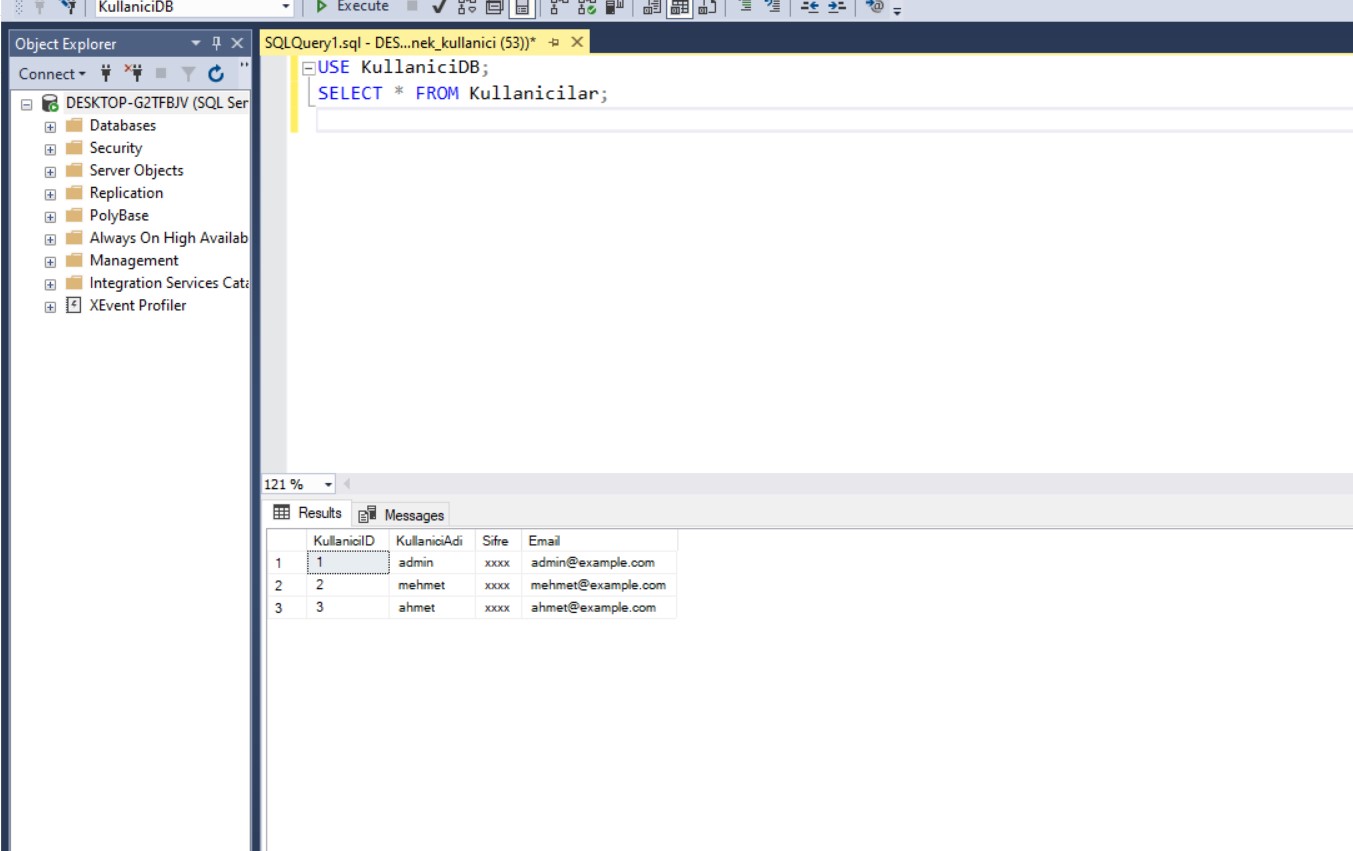
Burada injectiona karşı koruma gösterilmiştir.

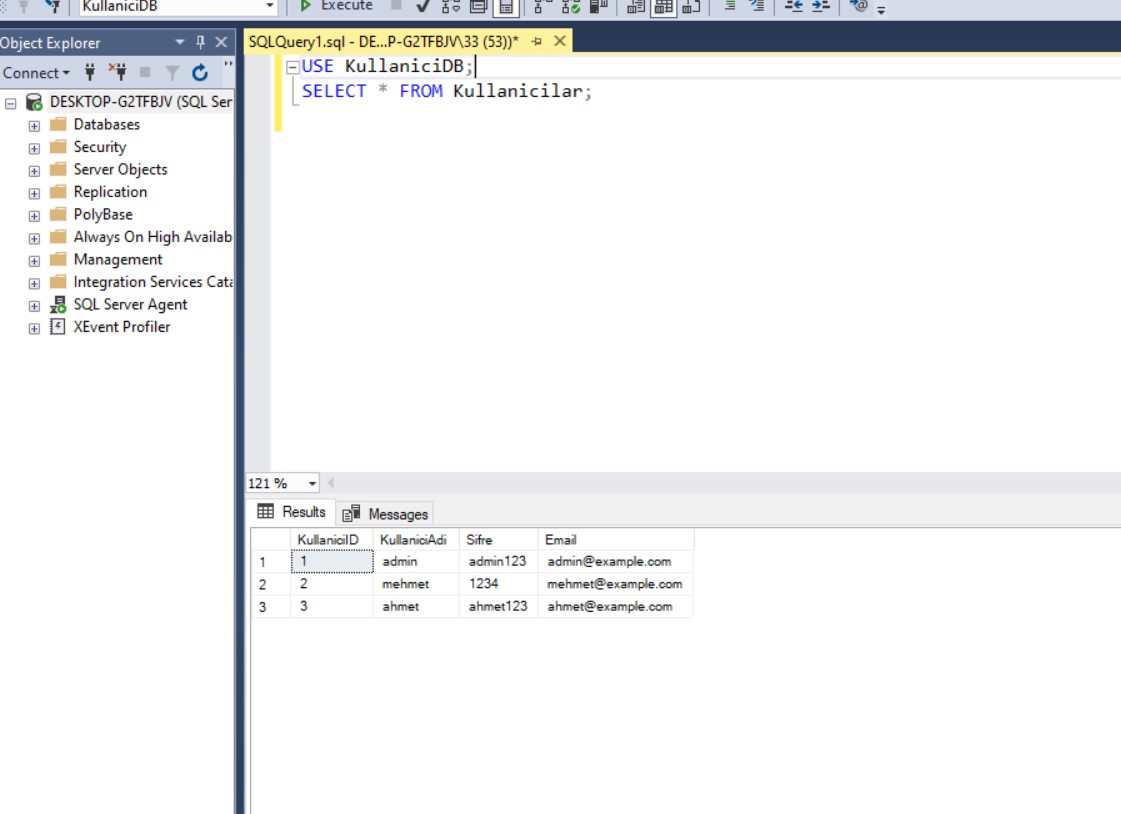
**Adım 4 – Verilerin Maskelemesi**

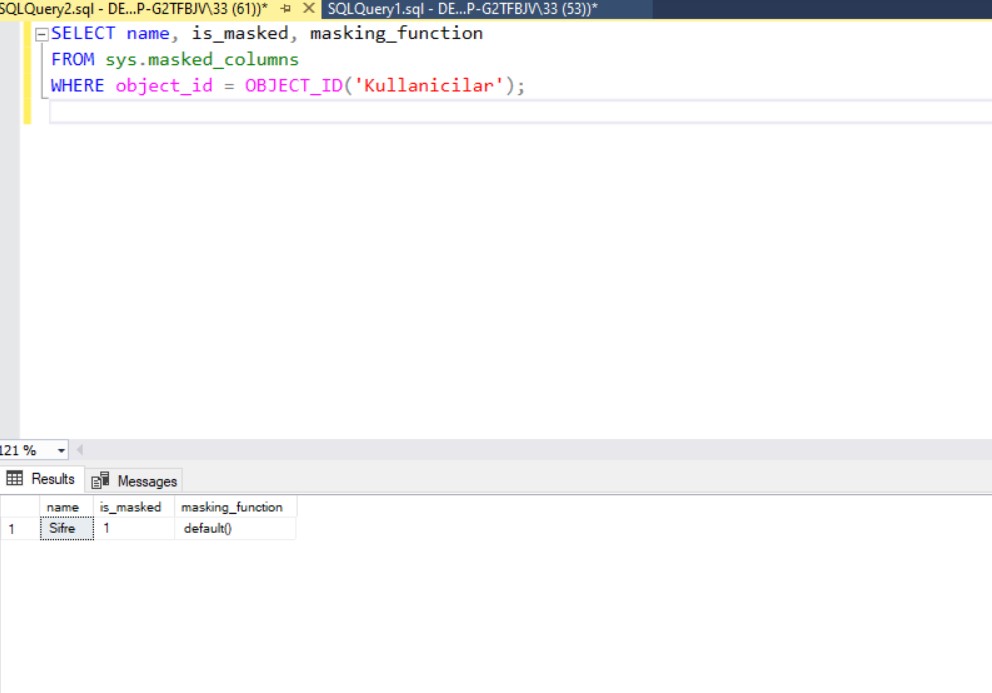
**Açıklama:**

Sifre kolonuna dinamik maskeleme uygulanarak yetkisiz kullanıcıların görememesi sağlanmıştır.





ornek\_kullanici ile bağlanıp Sifre kolonunun maskelenmiş haliYönetici ile aynı sorguda Sifre kolonunun görünmesi

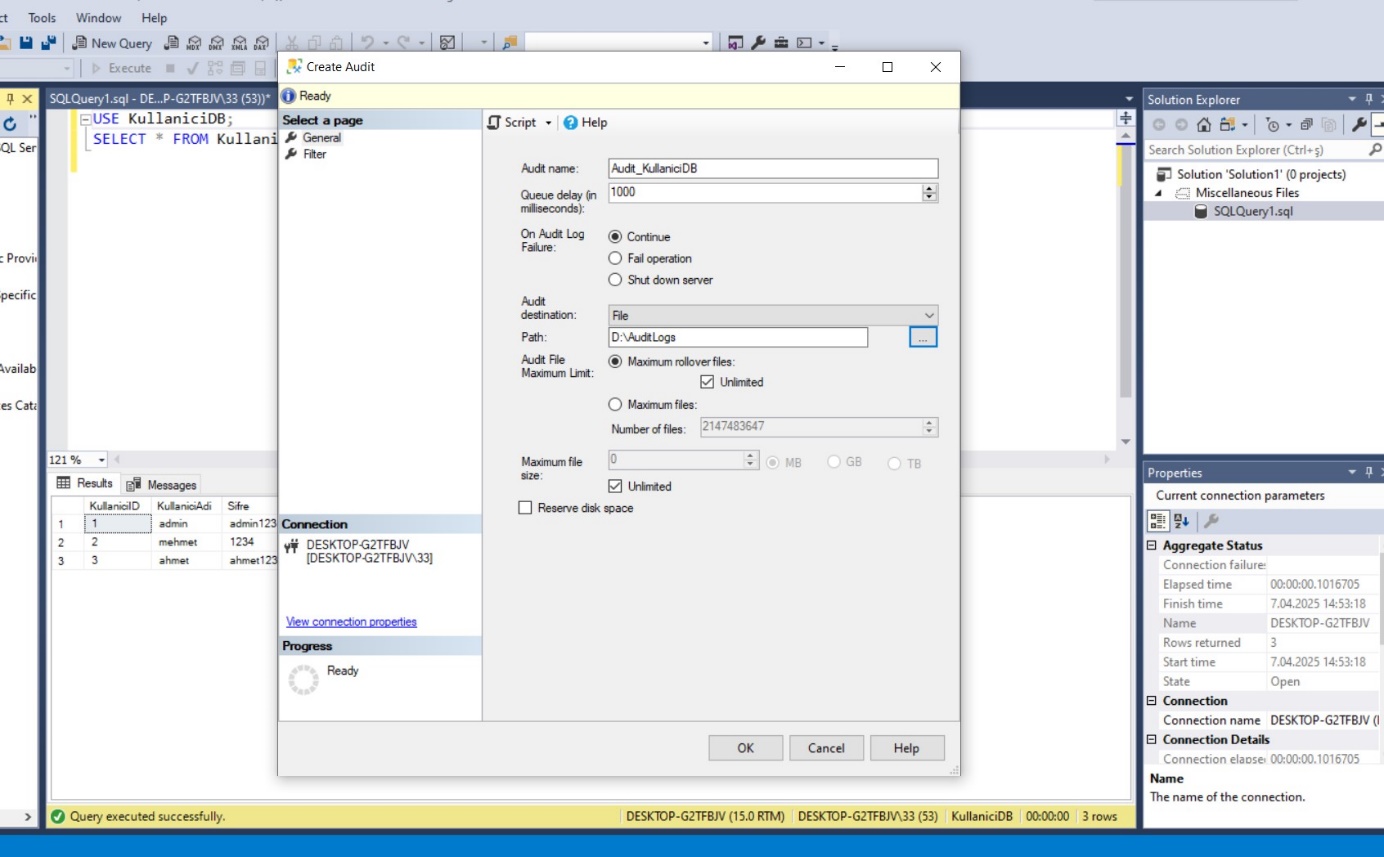


sys.masked\_columns ile kontrol ekranı

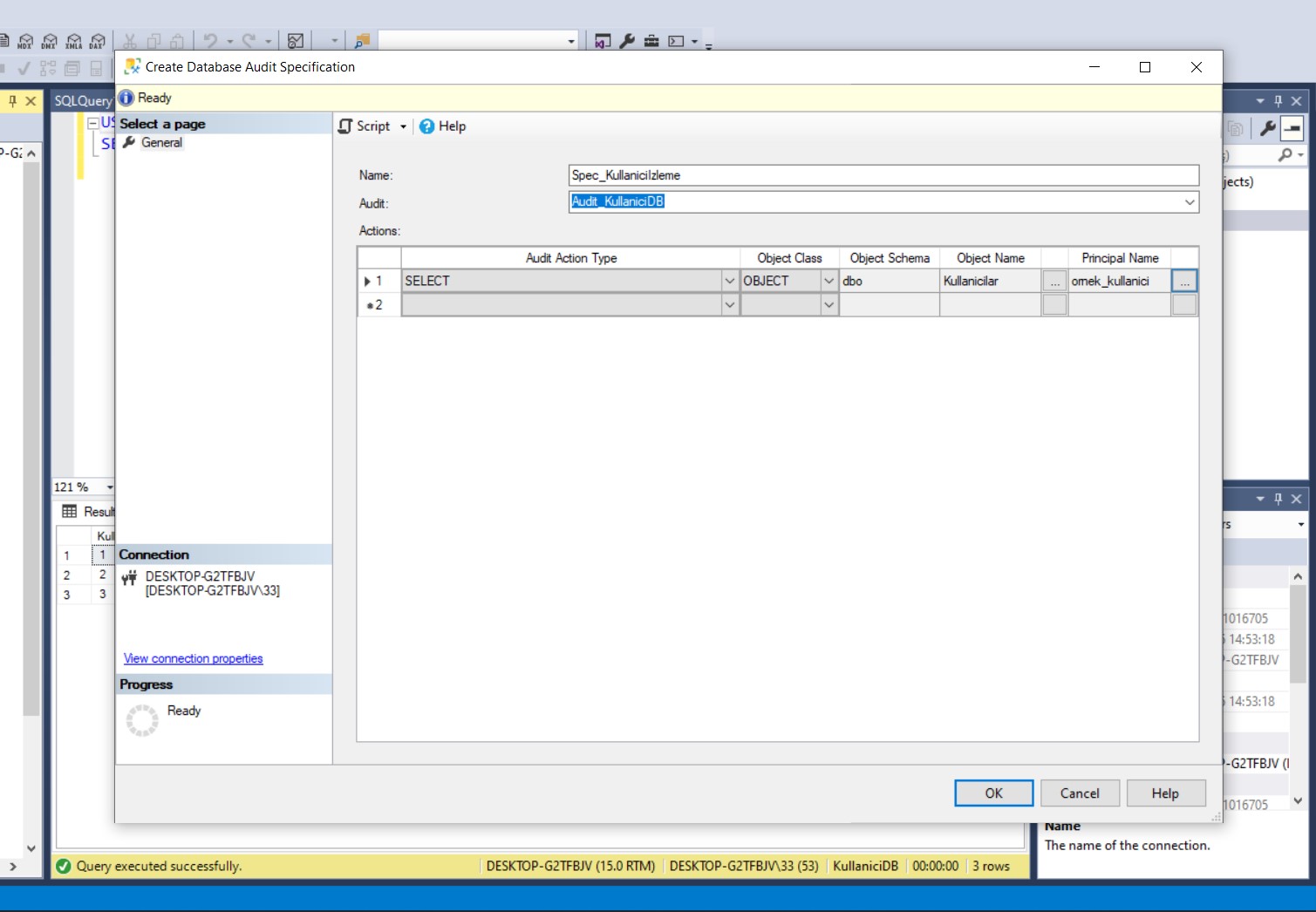
**Adım 5 – Audit: Kullanıcı İşlemlerini İzleme**

**Açıklama:**

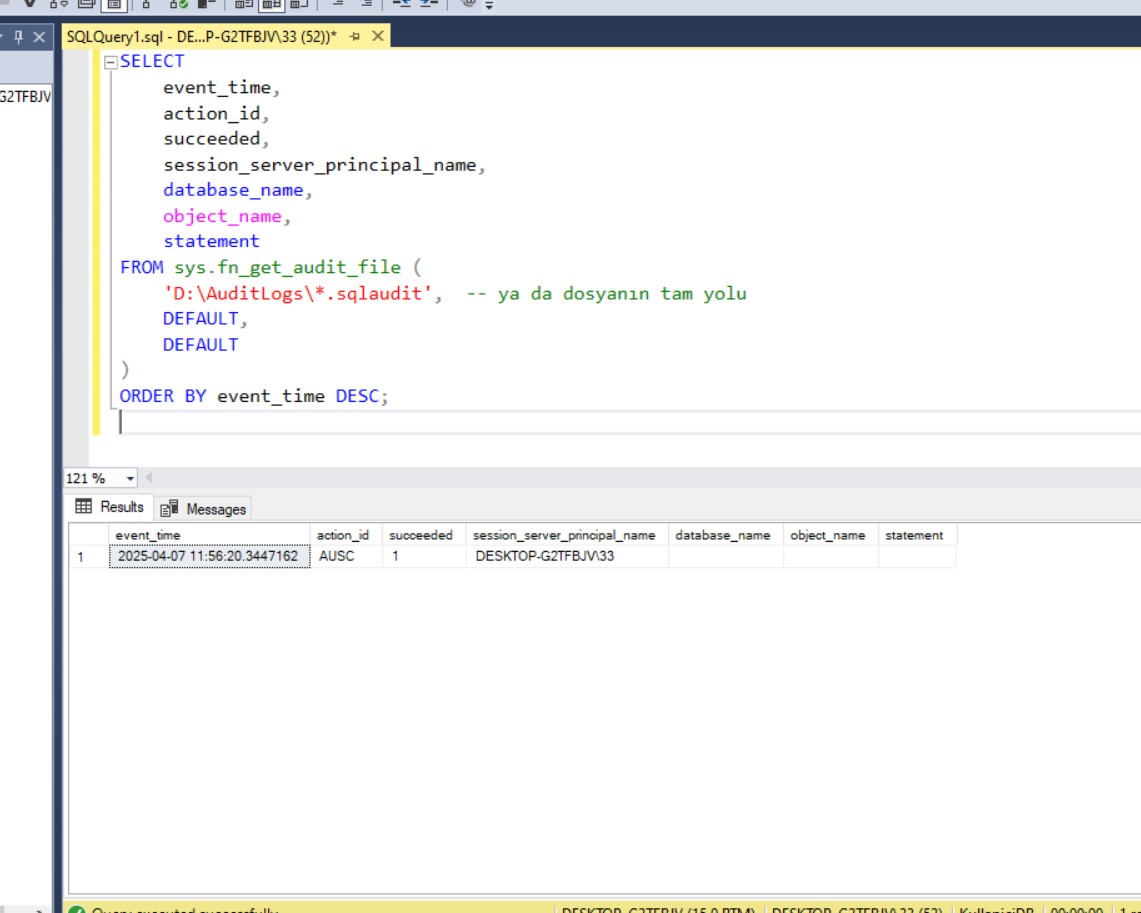
ornek\_kullanici tarafından yapılan SELECT sorguları SQL Server Audit sistemi ile izlenmiştir.

****

Audit\_KullaniciDB adlı audit tanımlandı



Database Audit Specification oluşturuldu

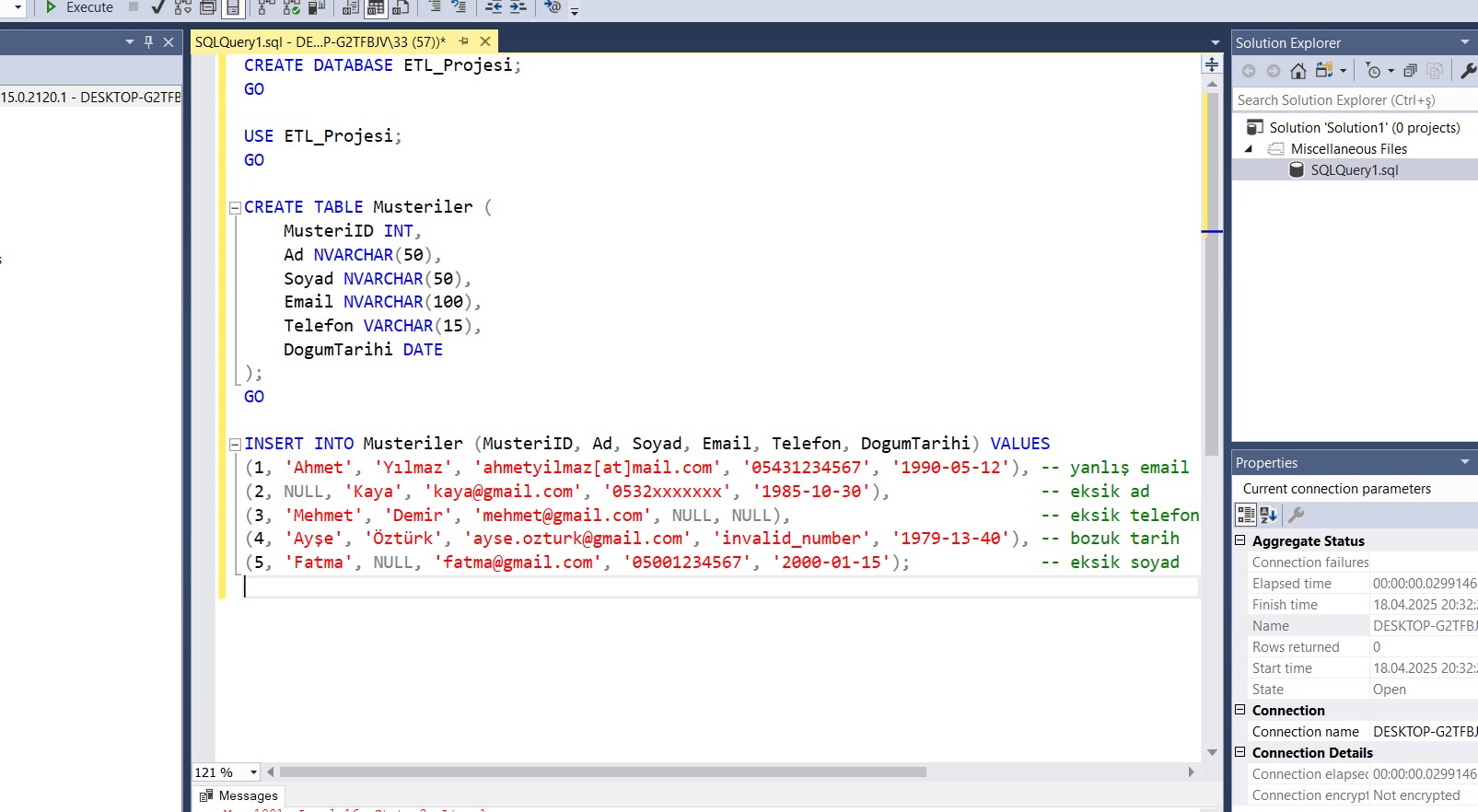


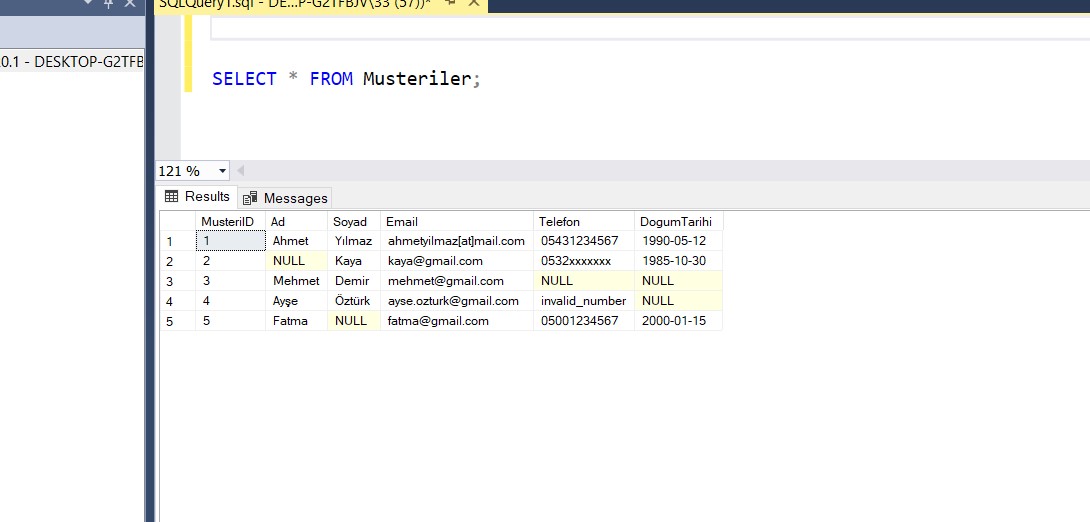
Kullanıcı sorgusundan sonra oluşan loglar

3) Veri Temizleme ve ETL Süreci Uygulaması

Adım 1: Hatalı Veri İçeren Kaynak Tabloyu Oluşturma

ETL\_Projesi adında yeni bir veritabanı oluşturulmuş ve Musteriler adında bir tablo tanımlanmıştır. Bu tabloya bilinçli olarak hatalı ve eksik veriler eklenmiştir.

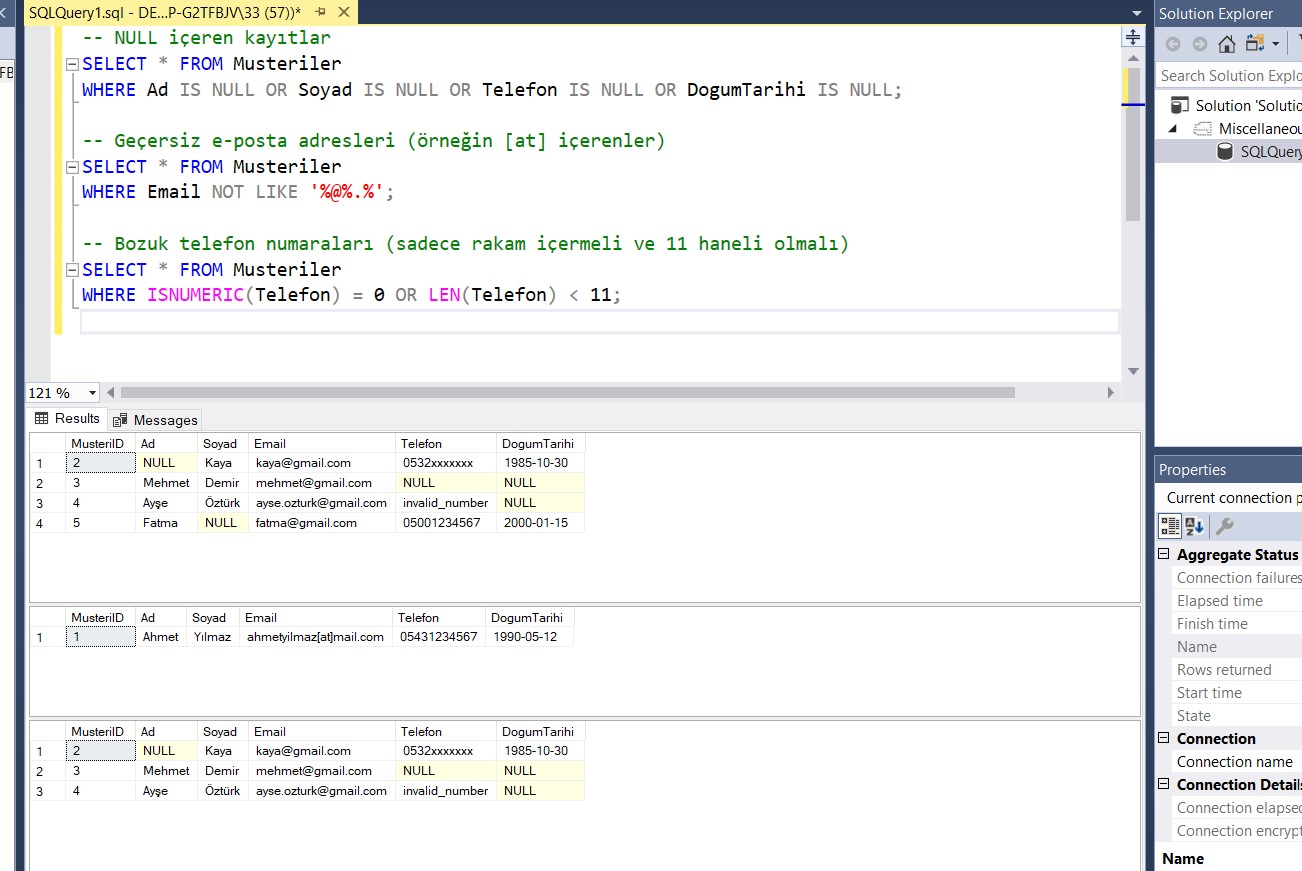


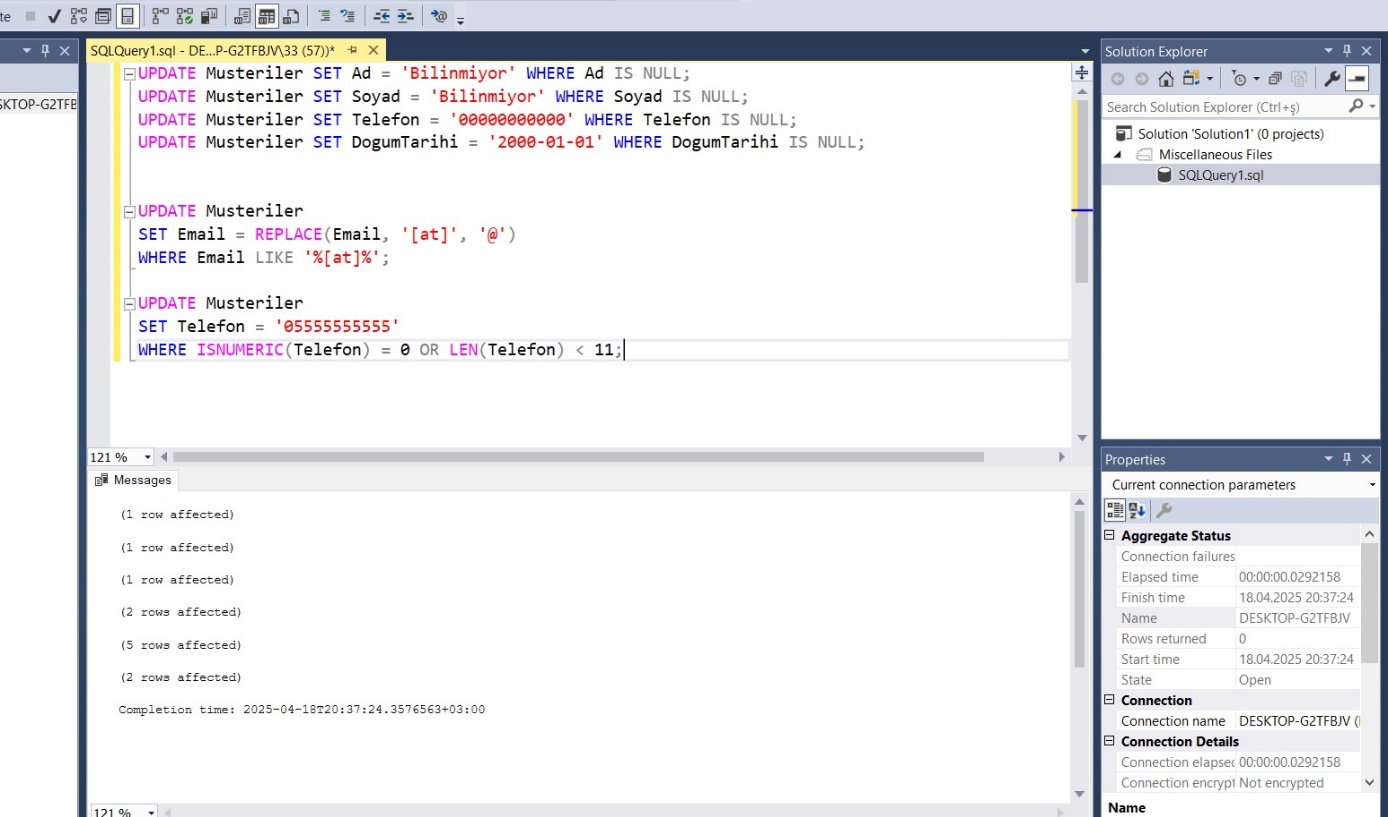


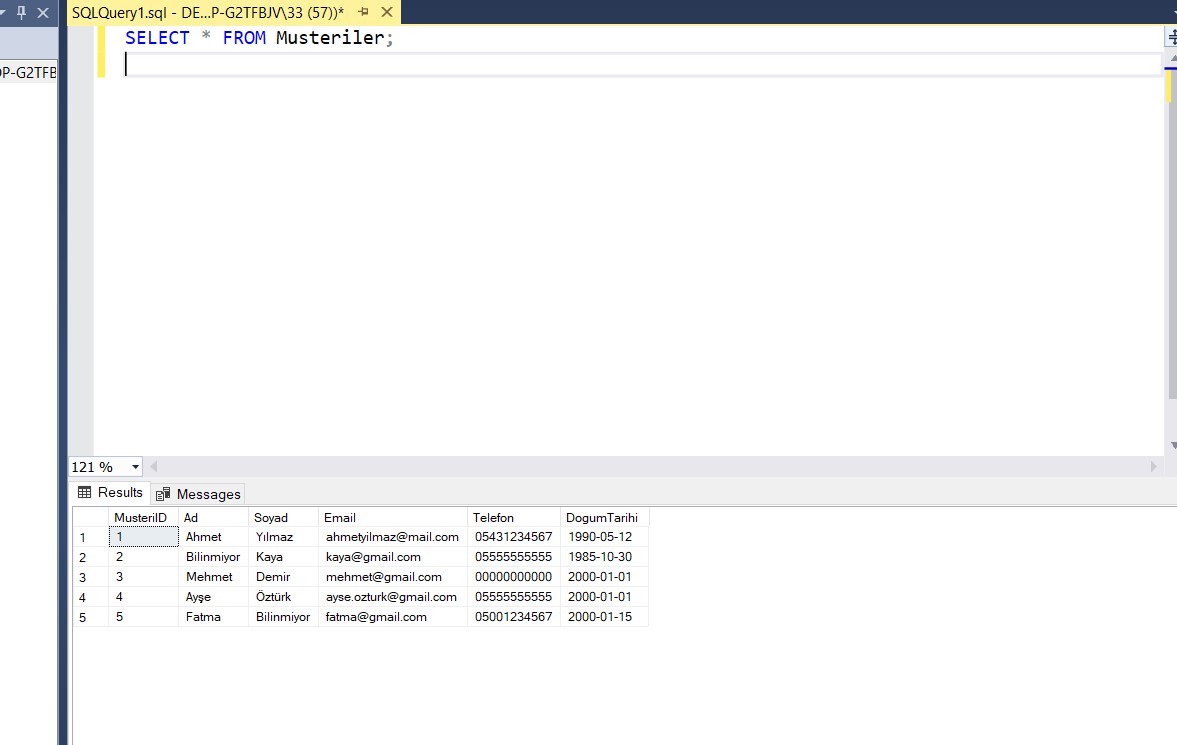
Adım 2: Veri Temizleme İşlemleri

Eksik veya hatalı veriler tespit edilip düzeltildi. Örneğin:

* NULL alanlar Bilinmiyor, 00000000000, 2000-01-01 gibi değerlerle güncellendi
* E-posta formatları düzeltildi ([at] → @)
* Bozuk telefon numaraları geçerli hale getirildi

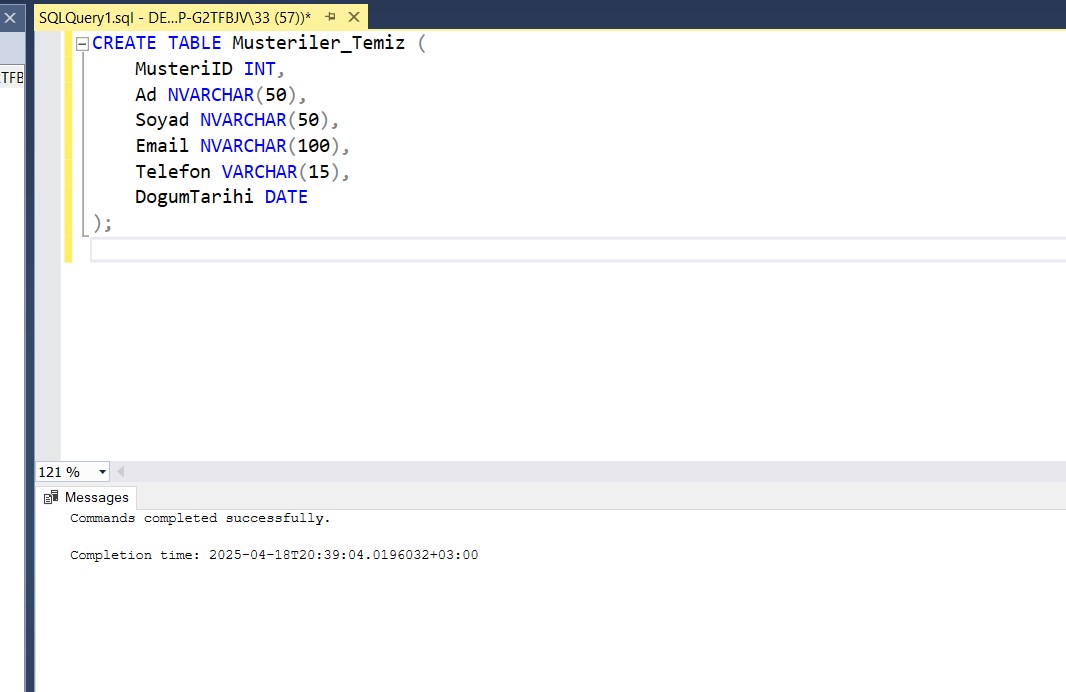




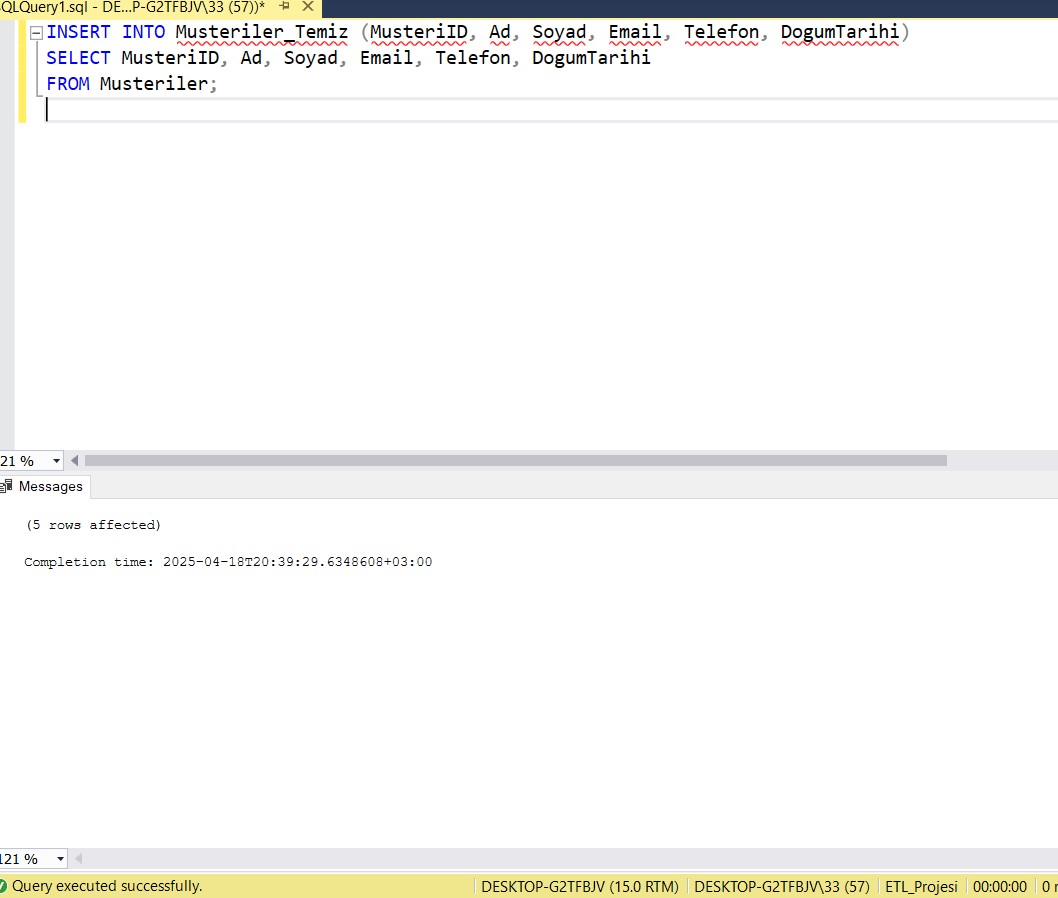


Adım 3: ETL - Temiz Veriyi Yeni Tabloya Aktarma (Load)

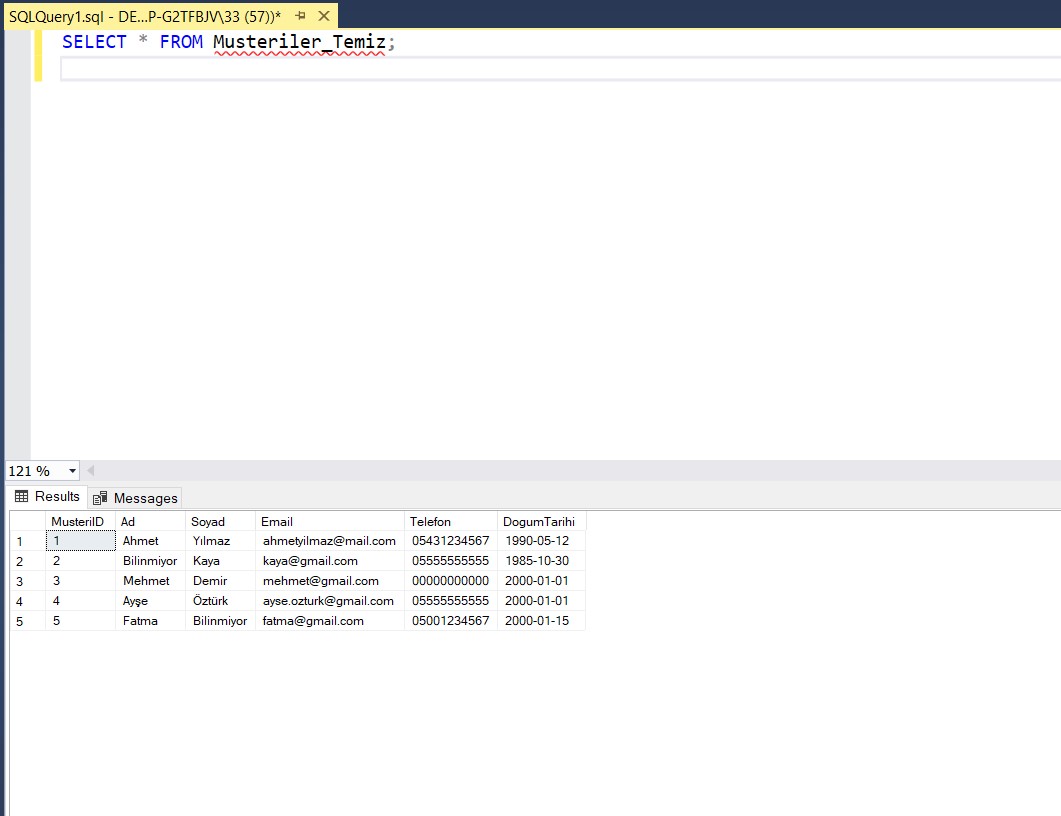
Temizlenmiş veriler Musteriler\_Temiz adlı yeni tabloya aktarılmıştır. Bu işlem ETL sürecinin "Load" adımını temsil eder.



Burada yeni bir tablo oluşturuldu.



Temiz veriyi bu yeni tablomuza aktardık.



Bu da yeni tablomuz.