

ANKARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4522 Veri Temizleme ve ETL Süreçleri Tasarımı

Furkan Yağcı

Görkem HAZAR

20290301

21290428

19.06.2025

Github : <https://github.com/FurkanYagciiii>

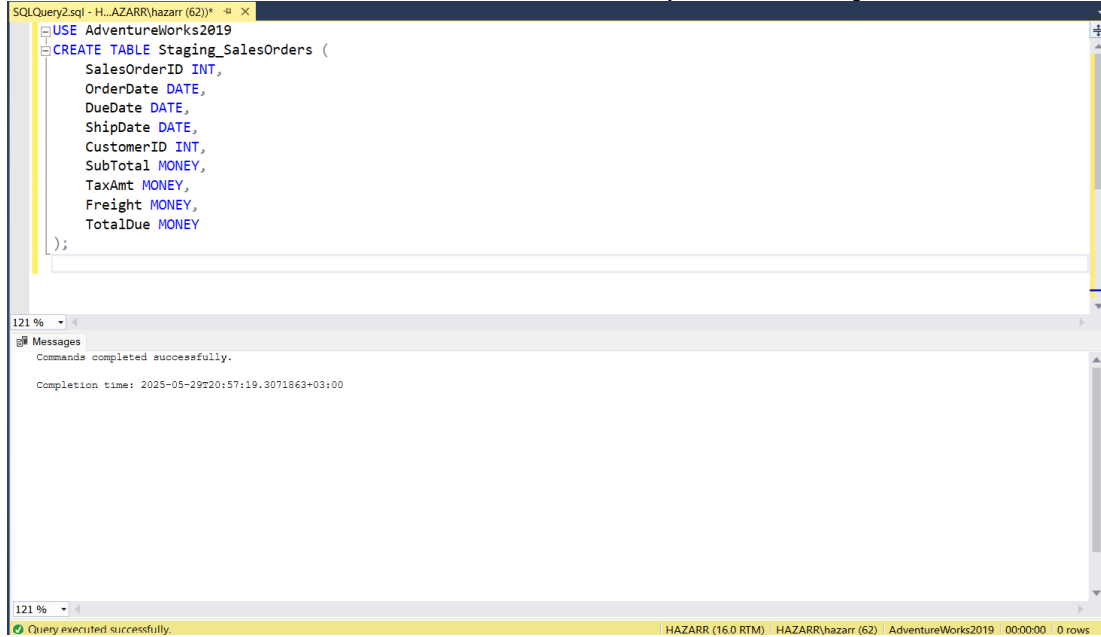
Video Linki : https://drive.google.com/file/d/1zwThwkri8isIH86FPsbm1rLsxr7-wps1/view?usp=drive_link

PROJE : Veri Temizleme ve ETL Süreçleri Tasarımı

ADIM 1 – VERİYİ ÇEK (Extract)

1.1. Staging Tablosu Oluşturuldu

Verileri geçici olarak tutmak için Staging_SalesOrders adlı tablo oluşturuldu. Raw verileri temizleme öncesi izole ortamda toplaması amaçlandı



```
USE AdventureWorks2019
CREATE TABLE Staging_SalesOrders (
    SalesOrderID INT,
    OrderDate DATE,
    DueDate DATE,
    ShipDate DATE,
    CustomerID INT,
    SubTotal MONEY,
    TaxAmt MONEY,
    Freight MONEY,
    TotalDue MONEY
);
```

Messages

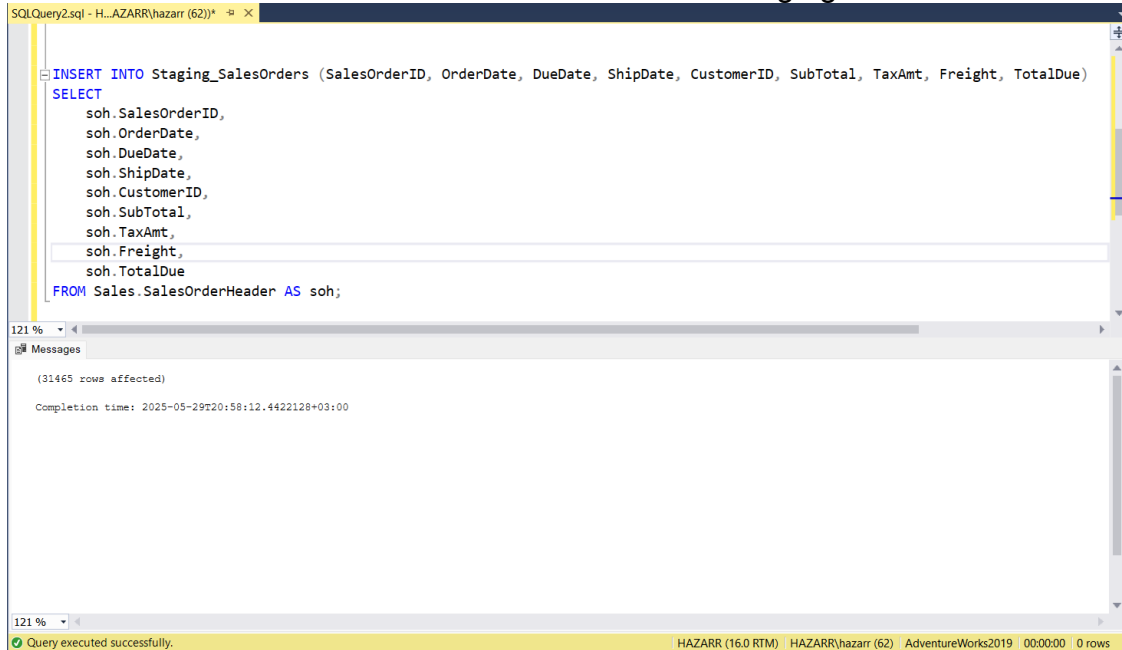
Commands completed successfully.

Completion time: 2025-05-29T20:57:19.3071863+03:00

Query executed successfully. HAZARR (16.0 RTM) HAZARR\hazarr (62) AdventureWorks2019 00:00:00 0 rows

1.2. Ham Veri Aktarıldı

Sales.SalesOrderHeader tablosundan 31.465 satır veri staging'e aktarıldı.



```
INSERT INTO Staging_SalesOrders (SalesOrderID, OrderDate, DueDate, ShipDate, CustomerID, SubTotal, TaxAmt, Freight, TotalDue)
SELECT
    soh.SalesOrderID,
    soh.OrderDate,
    soh.DueDate,
    soh.ShipDate,
    soh.CustomerID,
    soh.SubTotal,
    soh.TaxAmt,
    soh.Freight,
    soh.TotalDue
FROM Sales.SalesOrderHeader AS soh;
```

Messages

(31465 rows affected)

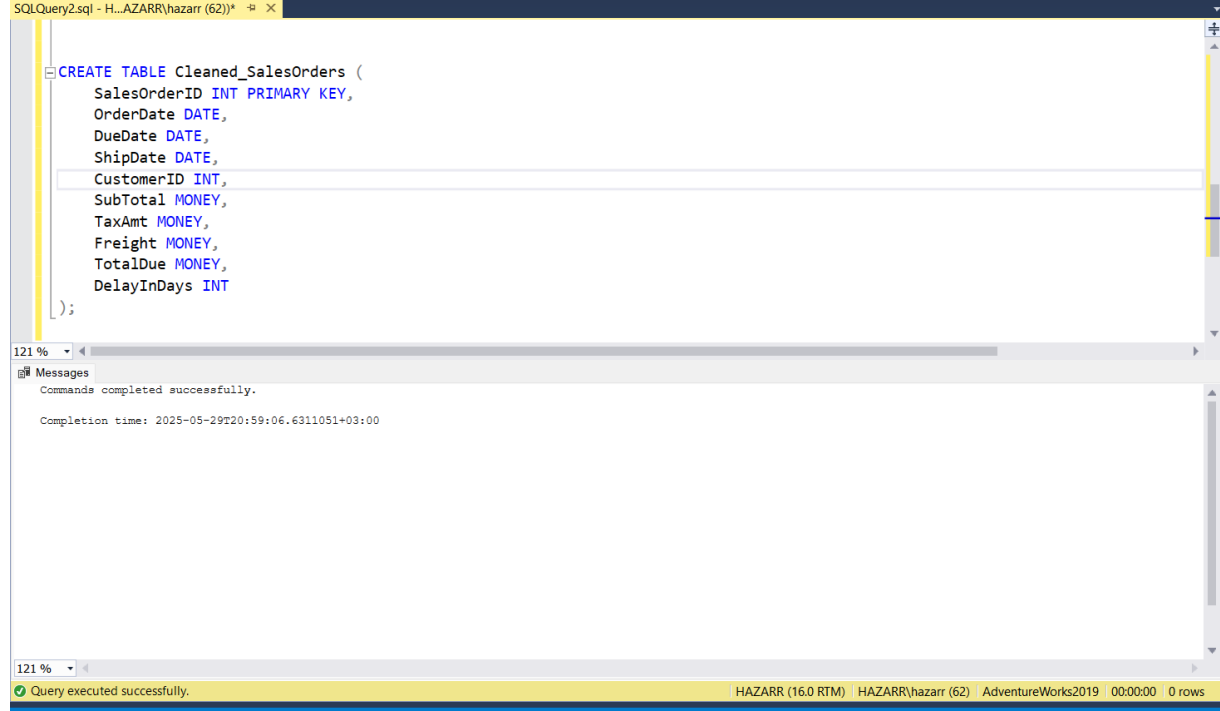
Completion time: 2025-05-29T20:58:12.4422128+03:00

Query executed successfully. HAZARR (16.0 RTM) HAZARR\hazarr (62) AdventureWorks2019 00:00:00 0 rows

ADIM 2 – VERİ TEMİZLİĞİ ve DÖNÜŞÜM (Transform)

2.1. Cleaned_SalesOrders Tablosu

Temiz veriler için yeni bir yapı oluşturuldu.
Ek olarak DelayInDays sütunu eklendi.

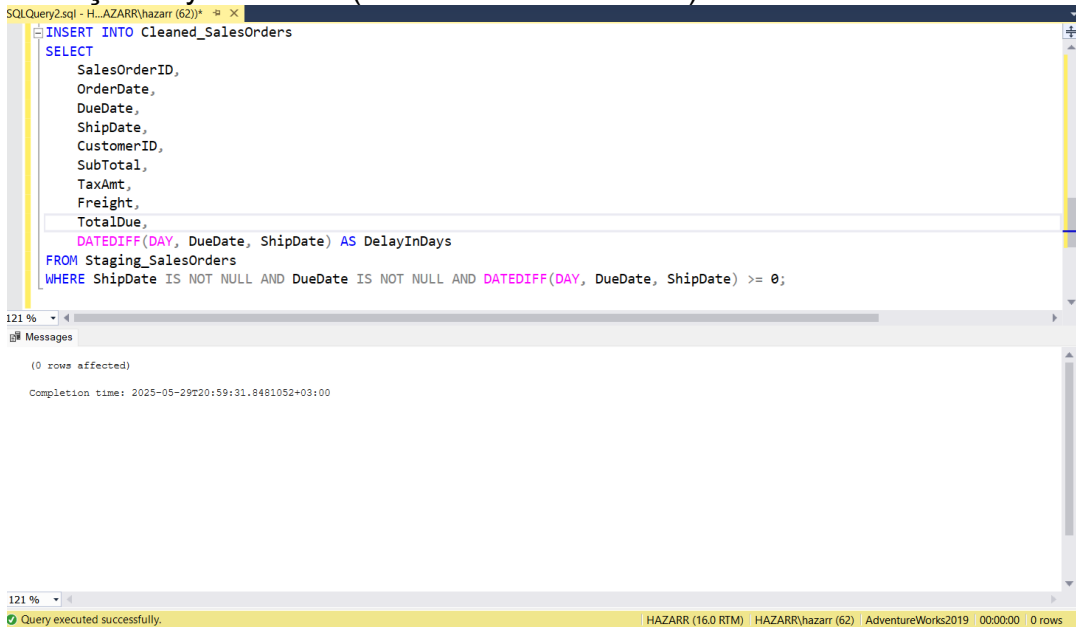


```
SQLQuery2.sql - H...AZARR\hazarr (62)*  
CREATE TABLE Cleaned_SalesOrders (  
    SalesOrderID INT PRIMARY KEY,  
    OrderDate DATE,  
    DueDate DATE,  
    ShipDate DATE,  
    CustomerID INT,  
    SubTotal MONEY,  
    TaxAmt MONEY,  
    Freight MONEY,  
    TotalDue MONEY,  
    DelayInDays INT  
);  
Messages  
Commands completed successfully.  
Completion time: 2025-05-29T20:59:06.6311051+03:00  
121 %  
Query executed successfully. HAZARR (16.0 RTM) HAZARR\hazarr (62) AdventureWorks2019 00:00:00 0 rows
```

2.2. Veri Temizliği ve Gecikme Hesabı

- ShipDate ve DueDate boş olmayan kayıtlar filtrelendi.
- Gecikme süresi (DelayInDays) hesaplandı.

Sonuç: 0 kayıt aktarıldı (Tüm veriler zamanında).

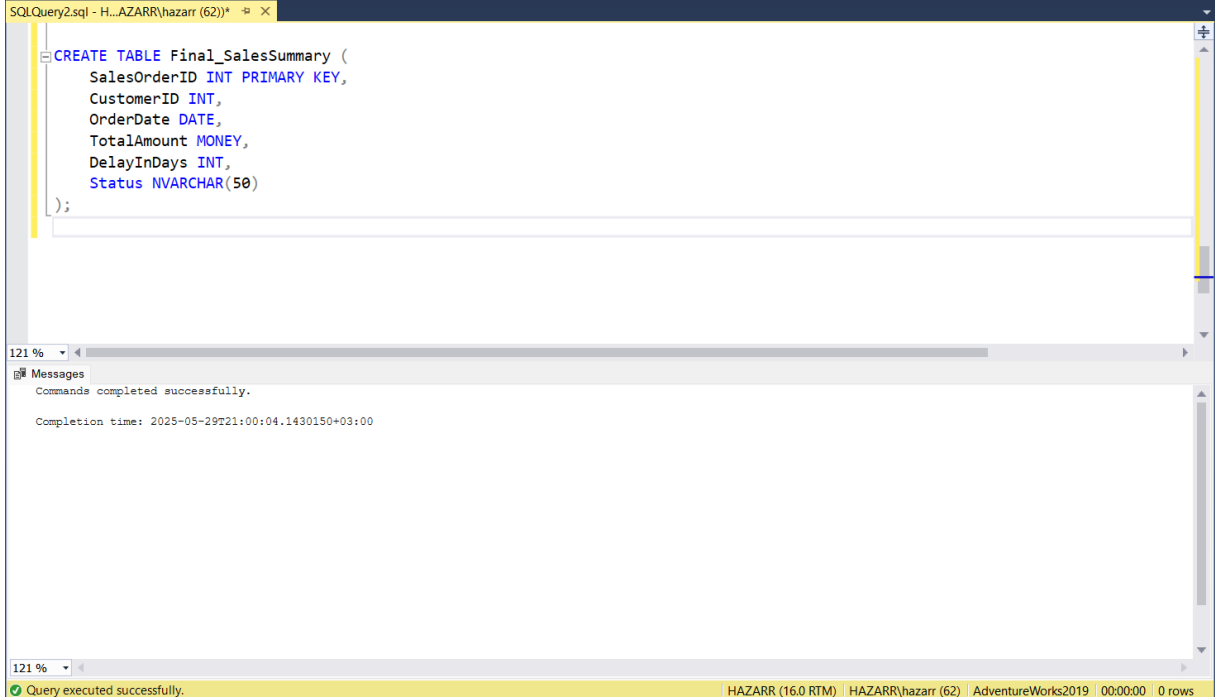


```
SQLQuery2.sql - H...AZARR\hazarr (62)*  
INSERT INTO Cleaned_SalesOrders  
SELECT  
    SalesOrderID,  
    OrderDate,  
    DueDate,  
    ShipDate,  
    CustomerID,  
    SubTotal,  
    TaxAmt,  
    Freight,  
    TotalDue,  
    DATEDIFF(DAY, DueDate, ShipDate) AS DelayInDays  
FROM Staging_SalesOrders  
WHERE ShipDate IS NOT NULL AND DueDate IS NOT NULL AND DATEDIFF(DAY, DueDate, ShipDate) >= 0;  
Messages  
(0 rows affected)  
Completion time: 2025-05-29T20:59:31.8481052+03:00  
121 %  
Query executed successfully. HAZARR (16.0 RTM) HAZARR\hazarr (62) AdventureWorks2019 00:00:00 0 rows
```

ADIM 3 – VERİYİ YÜKLE (Load)

3.1. Final Tablonun Oluşturulması

Raporlamaya uygun Final_SalesSummary tablosu tasarlandı.



```
CREATE TABLE Final_SalesSummary (  
    SalesOrderID INT PRIMARY KEY,  
    CustomerID INT,  
    OrderDate DATE,  
    TotalAmount MONEY,  
    DelayInDays INT,  
    Status NVARCHAR(50)  
);
```

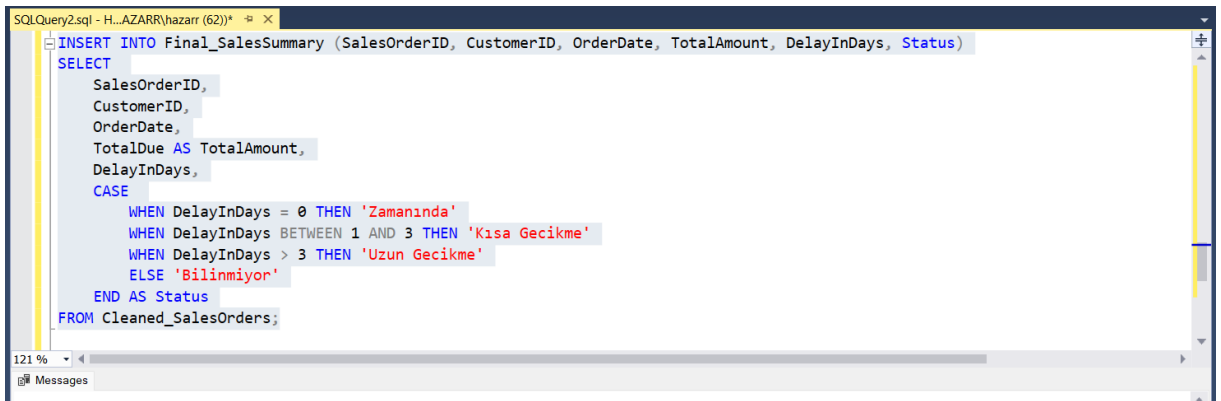
Messages
Commands completed successfully.
Completion time: 2025-05-29T21:00:04.1430150+03:00

Query executed successfully. HAZARR (16.0 RTM) HAZARR\hazarr (62) AdventureWorks2019 00:00:00 0 rows

3.2. Final Tablonun Doldurulması

Temizlenmiş veriler, gecikme sürelerine göre durum (Status) etiketleri ile yüklendi:

- Zamanında
- Kısa Gecikme
- Uzun Gecikme



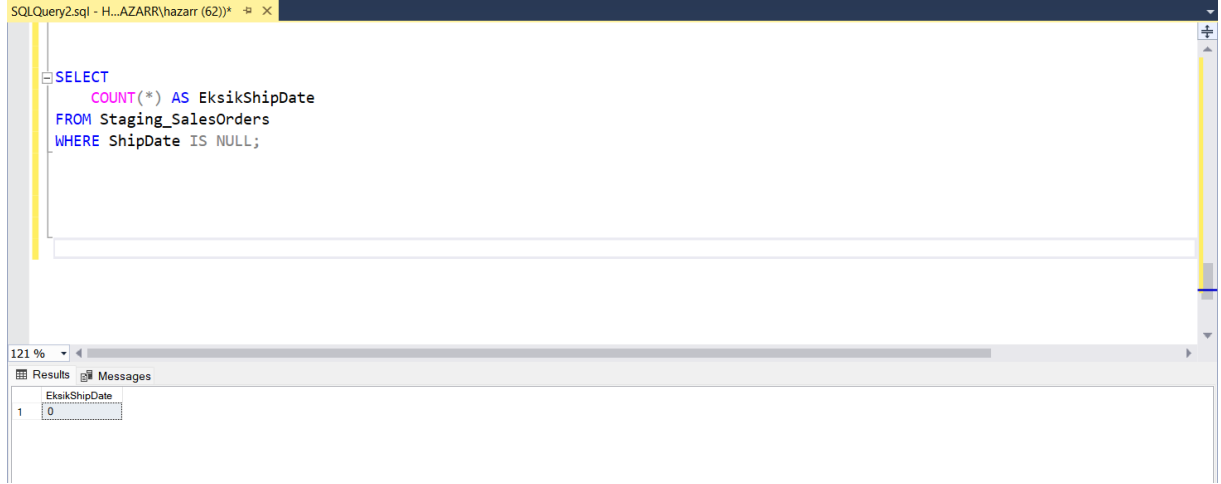
```
INSERT INTO Final_SalesSummary (SalesOrderID, CustomerID, OrderDate, TotalAmount, DelayInDays, Status)  
SELECT  
    SalesOrderID,  
    CustomerID,  
    OrderDate,  
    TotalDue AS TotalAmount,  
    DelayInDays,  
    CASE  
        WHEN DelayInDays = 0 THEN 'Zamanında'  
        WHEN DelayInDays BETWEEN 1 AND 3 THEN 'Kısa Gecikme'  
        WHEN DelayInDays > 3 THEN 'Uzun Gecikme'  
        ELSE 'Bilinmiyor'  
    END AS Status  
FROM Cleaned_SalesOrders;
```

Messages

ADIM 4 – VERİ KALİTESİ RAPORU

4.1. Eksik ShipDate Sayısı

Sonuç: 0 kayıt eksik.



The screenshot shows a SQL query window with the following query:

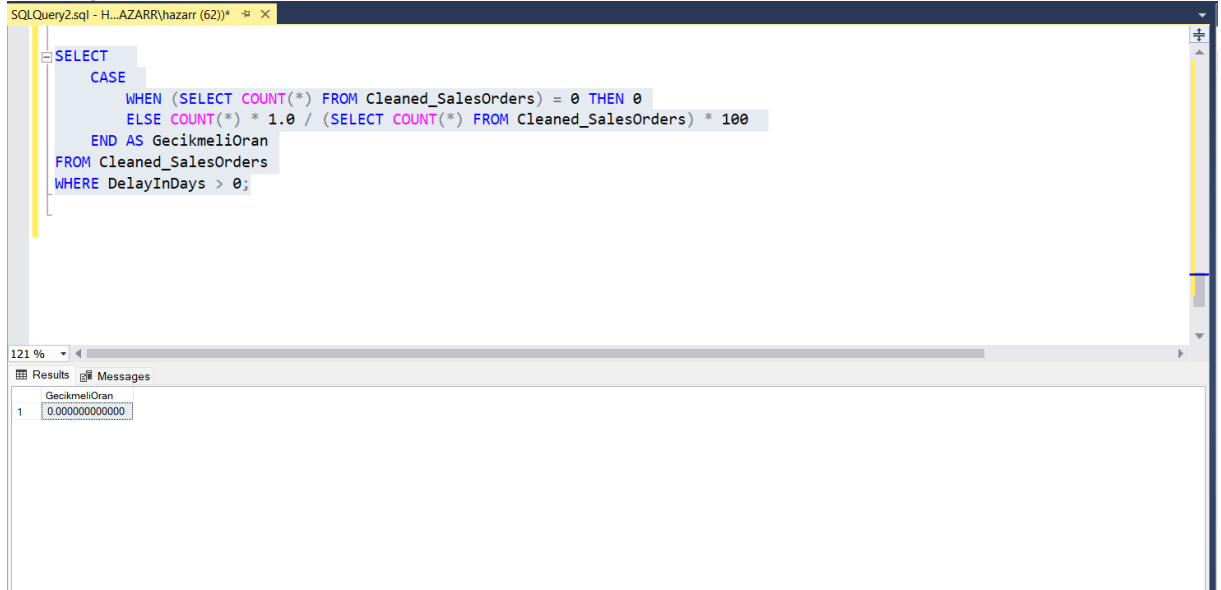
```
SELECT  
    COUNT(*) AS EksikShipDate  
FROM Staging_SalesOrders  
WHERE ShipDate IS NULL;
```

The results pane shows a single row with the value 0.

EksikShipDate
0

4.2. Gecikmeli Sipariş Oranı

Sonuç: %0 (Tüm siparişler zamanında teslim edilmiş).



The screenshot shows a SQL query window with the following query:

```
SELECT  
    CASE  
        WHEN (SELECT COUNT(*) FROM Cleaned_SalesOrders) = 0 THEN 0  
        ELSE COUNT(*) * 1.0 / (SELECT COUNT(*) FROM Cleaned_SalesOrders) * 100  
    END AS GecikmeliOran  
FROM Cleaned_SalesOrders  
WHERE DelayInDays > 0;
```

The results pane shows a single row with the value 0.000000000000.

GecikmeliOran
0.000000000000

4.3. Gecikme Türü İstatistiği

Sonuç: Final tabloda henüz durum dağılımı oluşmamış.



Proje gereklilikleri gösterilen tüm adımlarda yerine getirilmiş ve başarıyla proje tamamlanmıştır.