

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4522 PROJE RAPORU

Veri Tabanı Yükseltme ve Sürüm Yönetimi

<https://github.com/JineenRihawi>

Umut Akylbek kyzy 21291003

Cenin Rihavi 21291007

Öğretim Görevlisi Enver Bağcı

23.04.2025

İçindekiler

- 1. GİRİŞ**
- 2. Veritabanı Oluşturma**
 - 2.1. Veri tabanı Oluşturma Komutları
 - 2.2. Tabloları Oluşturma Komutu
- 3. Tablolara Veri Ekleme**
- 4. Okuma Yetkisine Sahip Login Oluşturulması**
- 5. Temel Sorgular**
- 6. Performans ve İndex Analizleri**
- 7. Şema Değişikliklerini İzleme**
- 8. Örnek Şema Değişikliği**
- 9. Log Versiyon Kontrolleri**
- 10. Kaynakça**

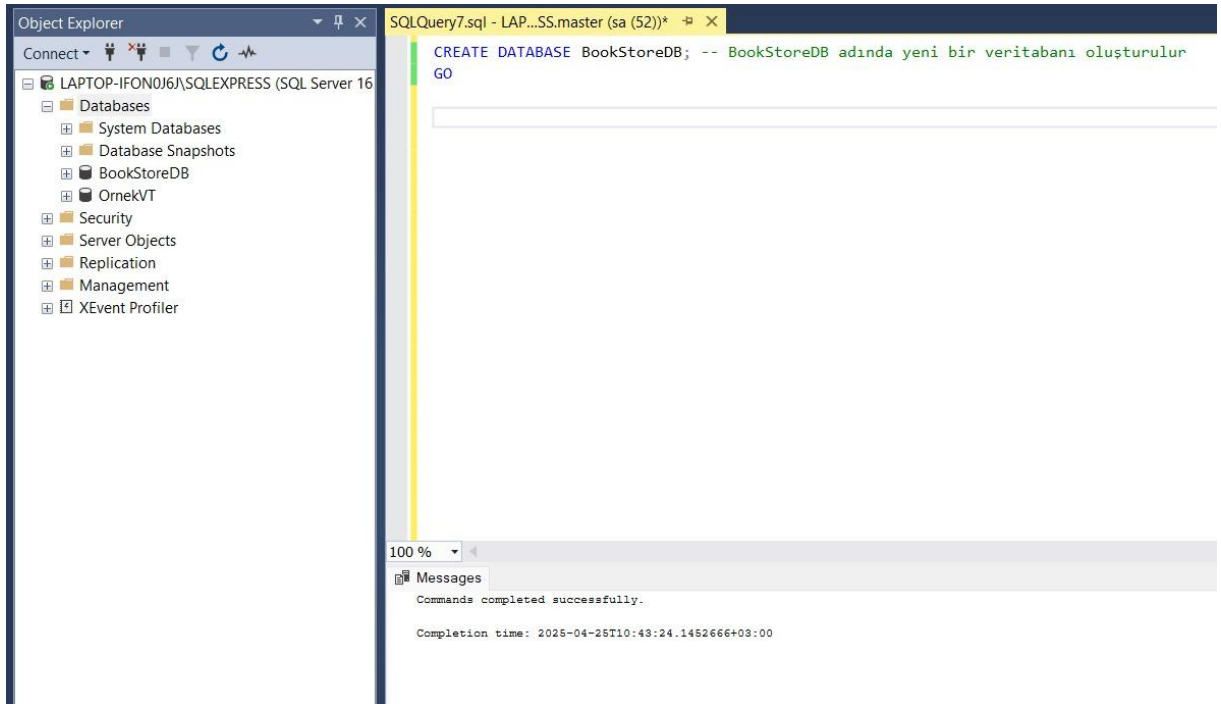
1. Giriş

Günümüzün hızla gelişen yazılım ve veri yönetim dünyasında, veritabanlarının güncel, güvenli ve sürdürülebilir olması, kurumların bilgi sistemlerinin verimliliği açısından kritik öneme sahiptir. Zaman içinde kullanılan veritabanı yönetim sistemlerinin yeni sürümleri yayımlandıkça, performans artışları, güvenlik iyileştirmeleri ve yeni özelliklerden yararlanmak amacıyla veritabanı sistemlerinin yükseltilmesi kaçınılmaz hale gelmektedir.

Bu rapor, mevcut bir veritabanının daha yeni bir sürüme nasıl yükseltileceğini, bu süreçte izlenecek stratejileri ve karşılaşılabilecek risklere karşı alınacak önlemleri detaylandırmaktadır. Ayrıca veritabanı yapısında yapılan değişikliklerin nasıl izleneceği, sürüm kontrolünün nasıl sağlanacağı ve yükseltme sonrası yapılacak testler ile olası bir başarısızlık durumunda uygulanacak geri dönüş planları ele alınacaktır. Bu kapsamda, veritabanı yükseltme ve sürüm yönetimi süreçlerinin etkin bir şekilde planlanması ve uygulanması, sistem sürekliliği ve veri bütünlüğü açısından büyük önem taşımaktadır.

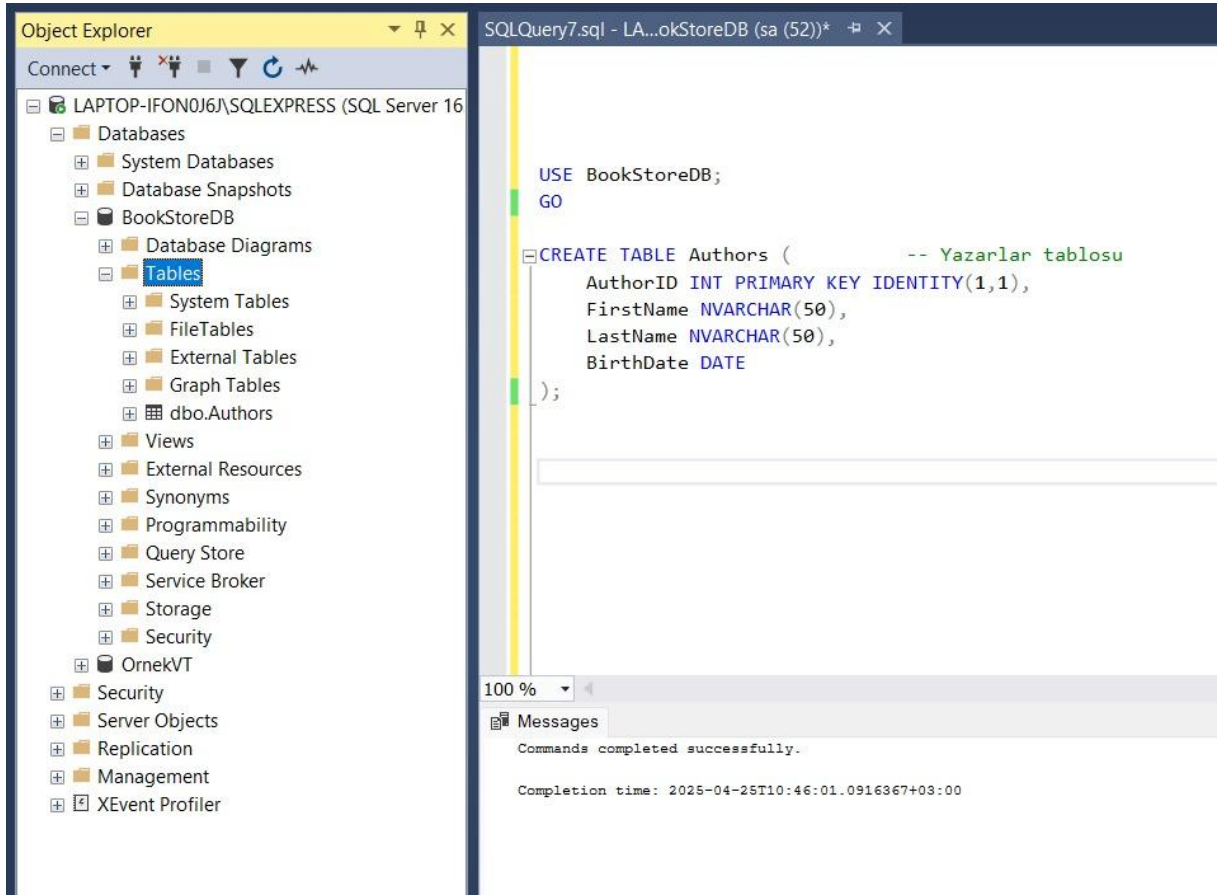
2. ÖRNEK VERİ TABANI OLUŞTURMA

2.1 Veri Tabanı Oluşturma Komutu:

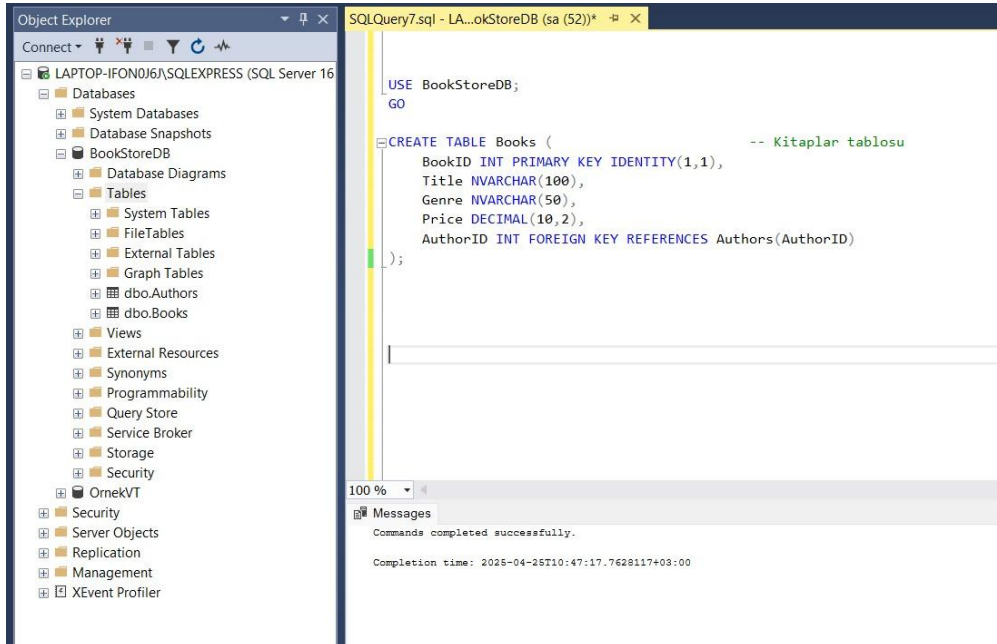


Şekil 1 Veri Tabanı Oluşturma

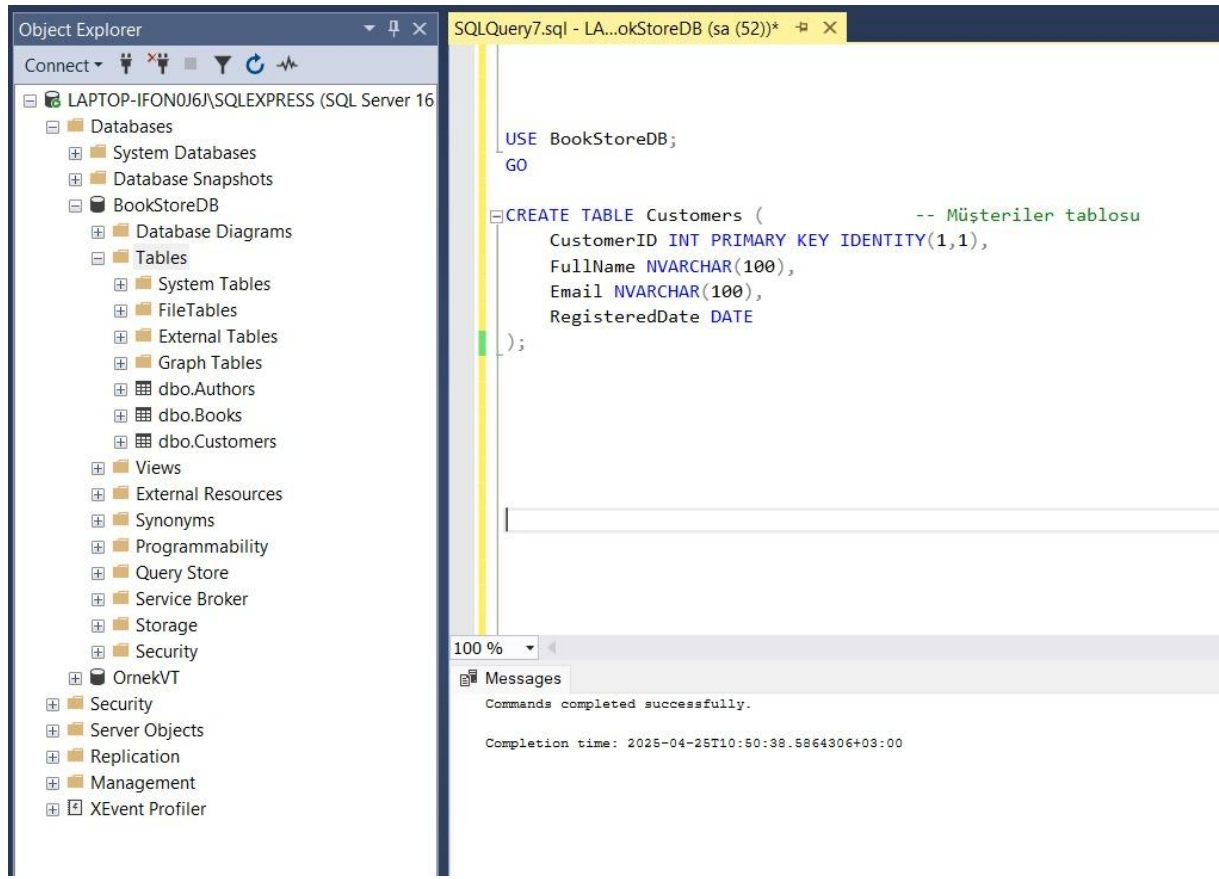
2.2 Tabloları Oluşturma Komutları:



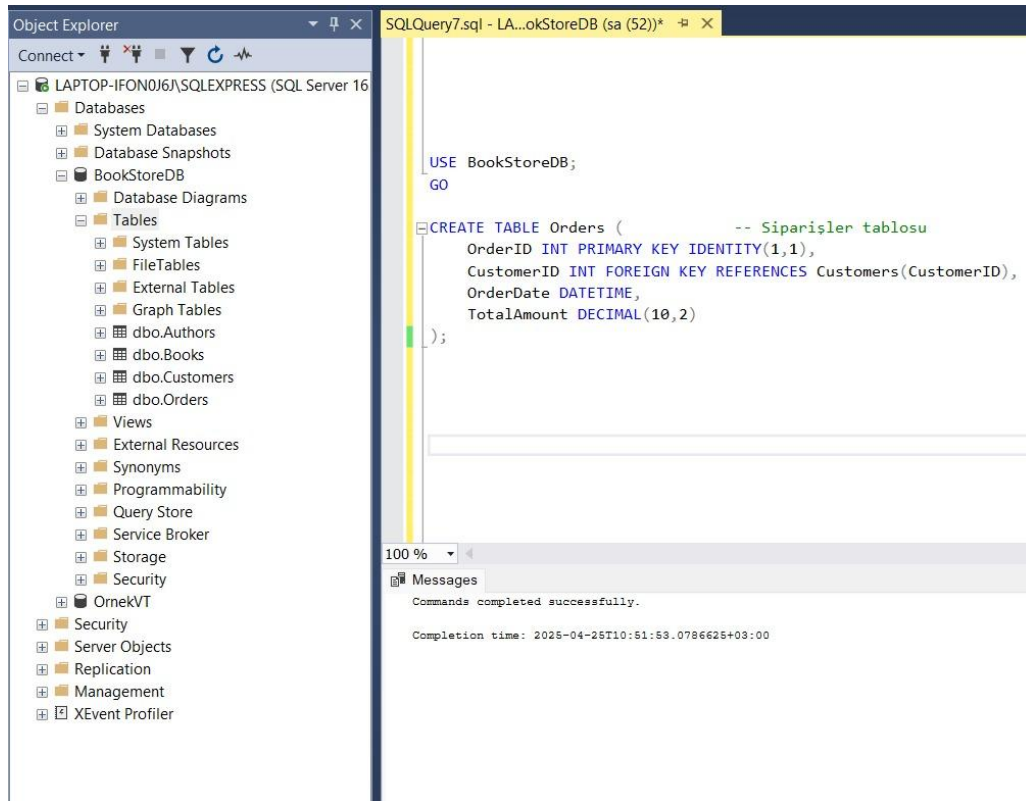
Şekil 2 Yazarlar Tablosu



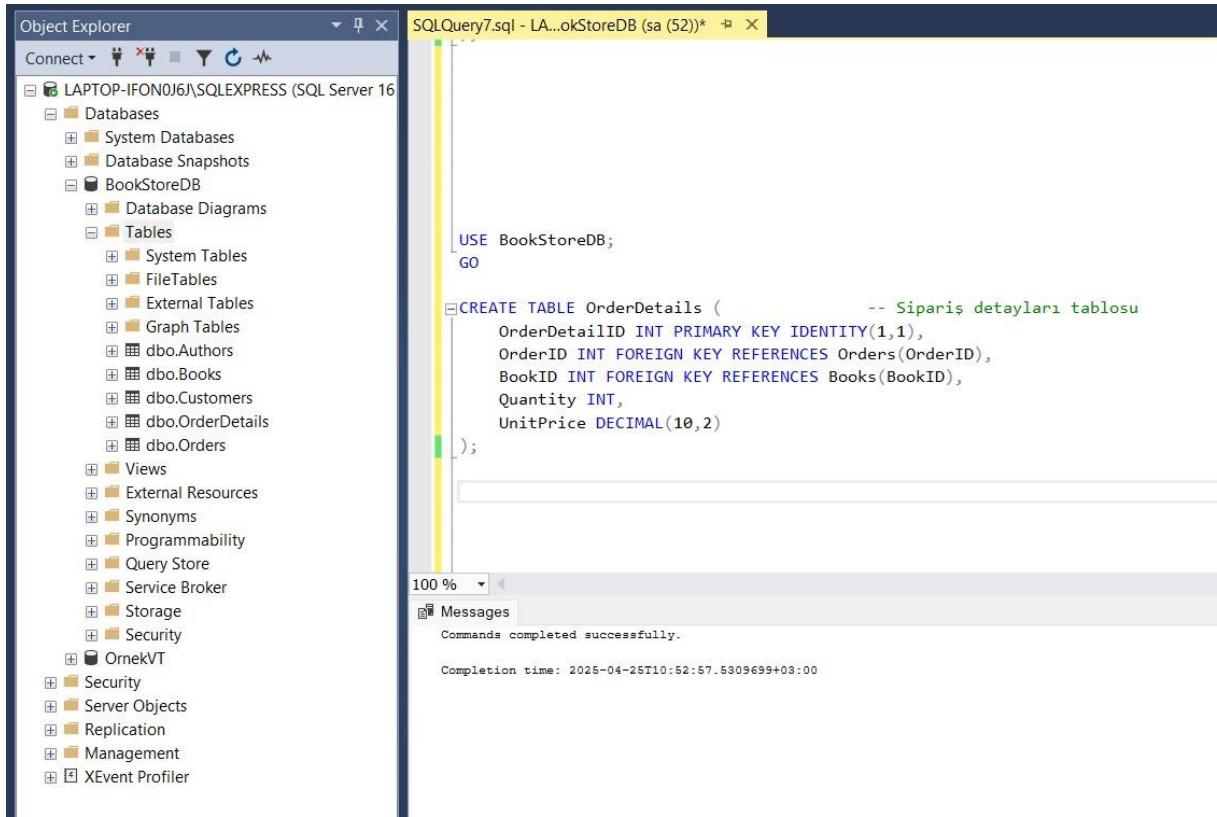
Şekil 3 Kitaplar Tablosu



Şekil 4 Müşteriler Tablosu

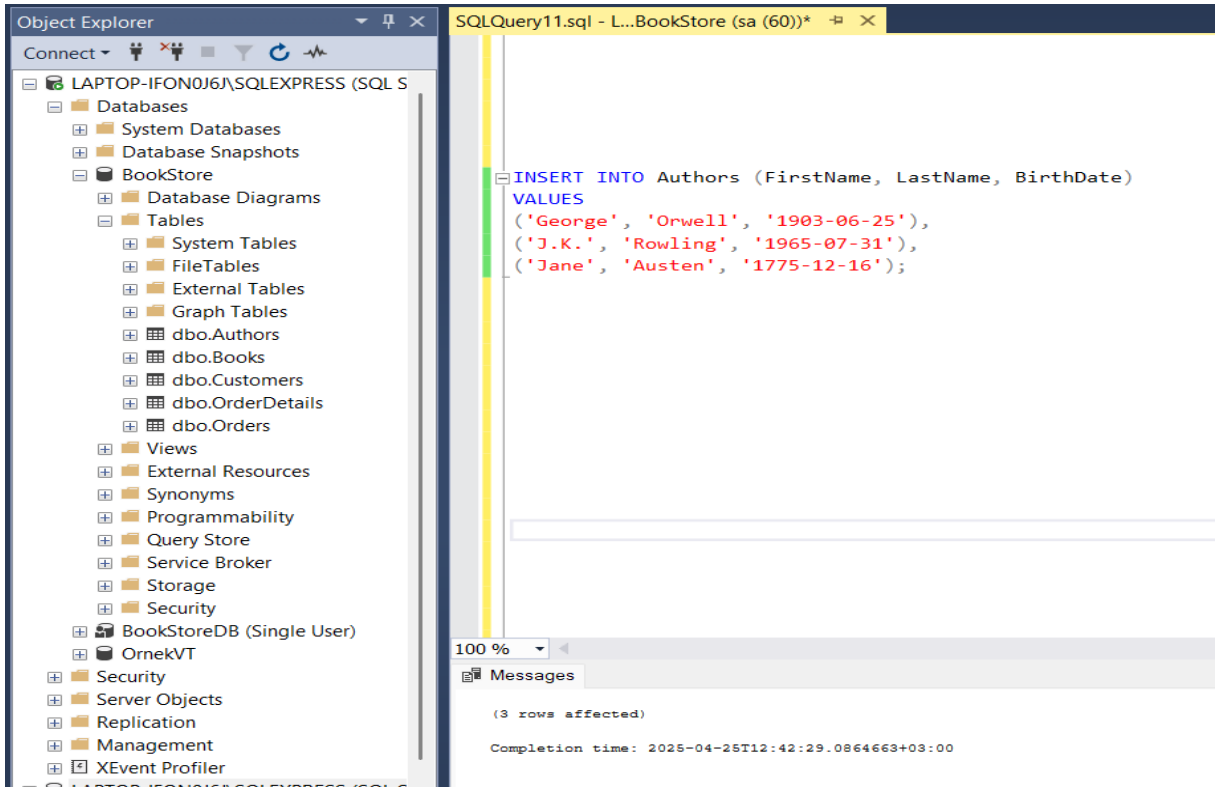


Şekil 5 Siparişler Tablosu

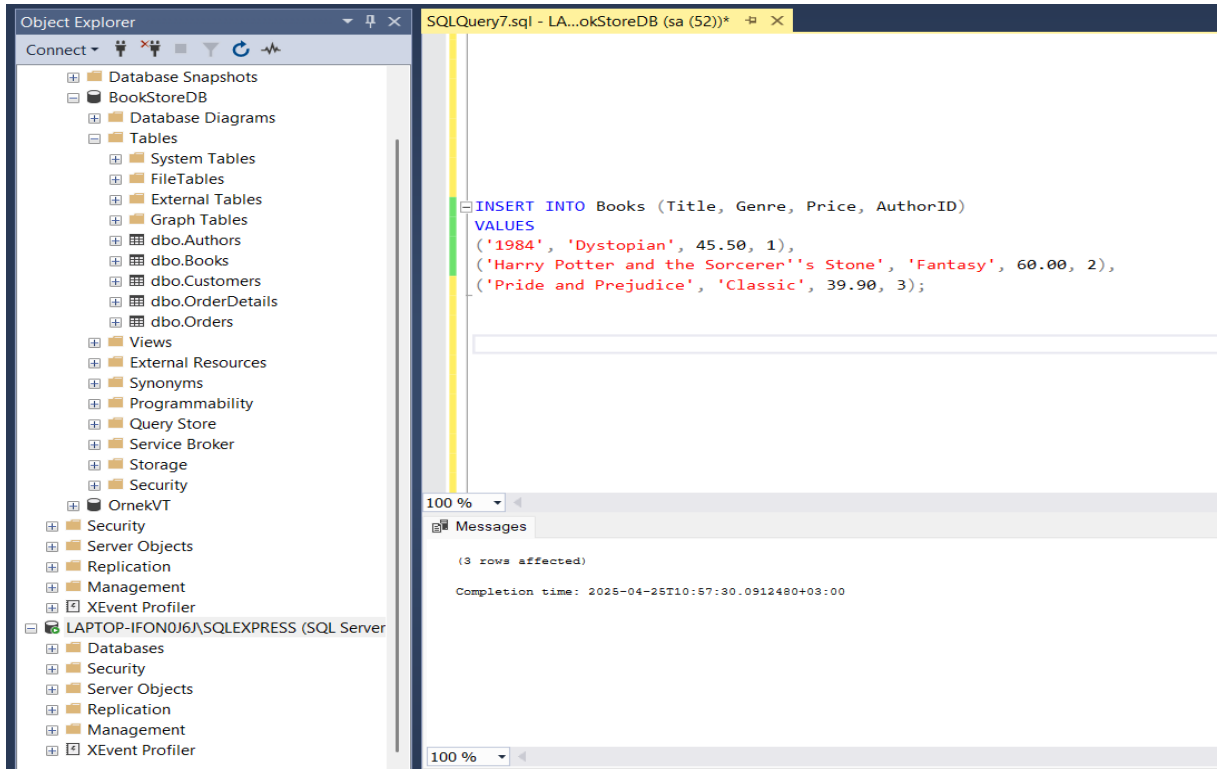


Şekil 5 Sipariş Detay Tablosu

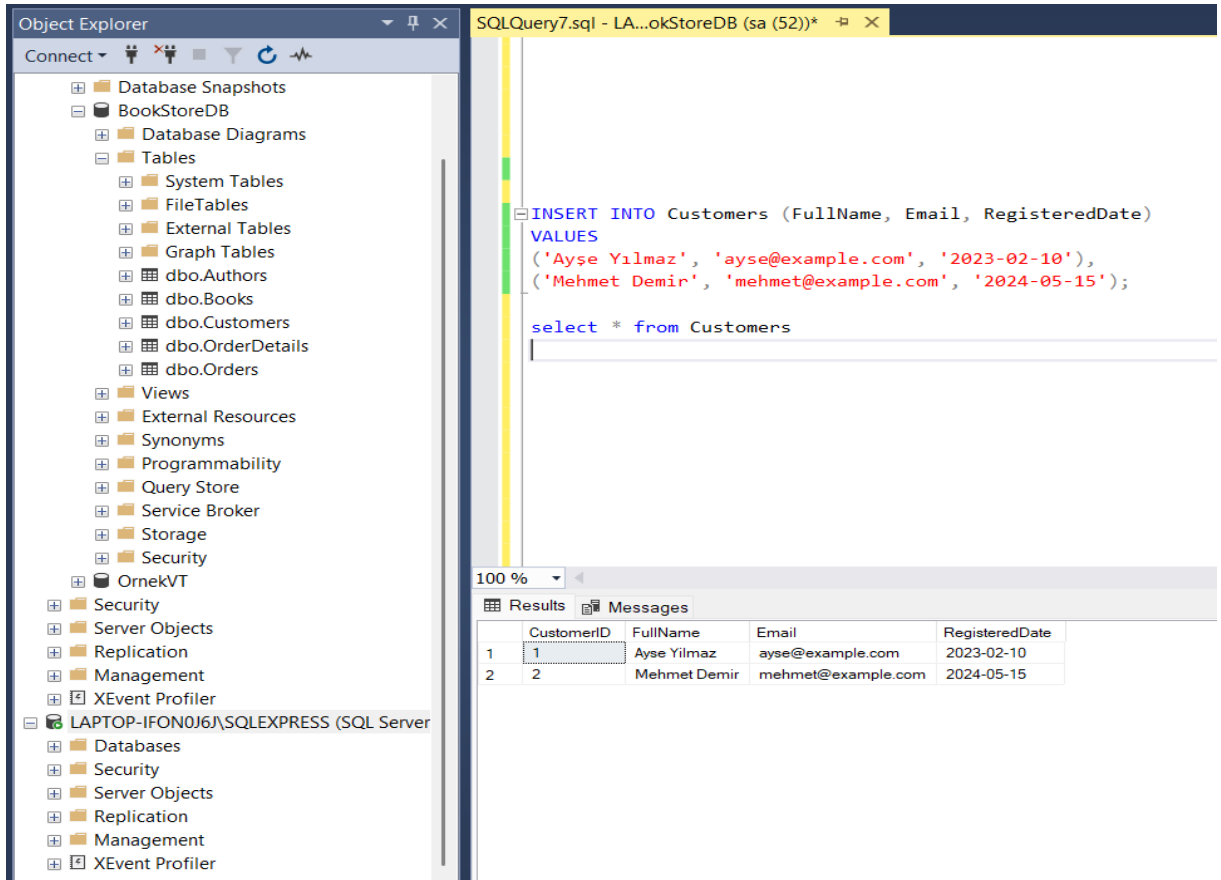
3. Tablolara Veri Ekleme



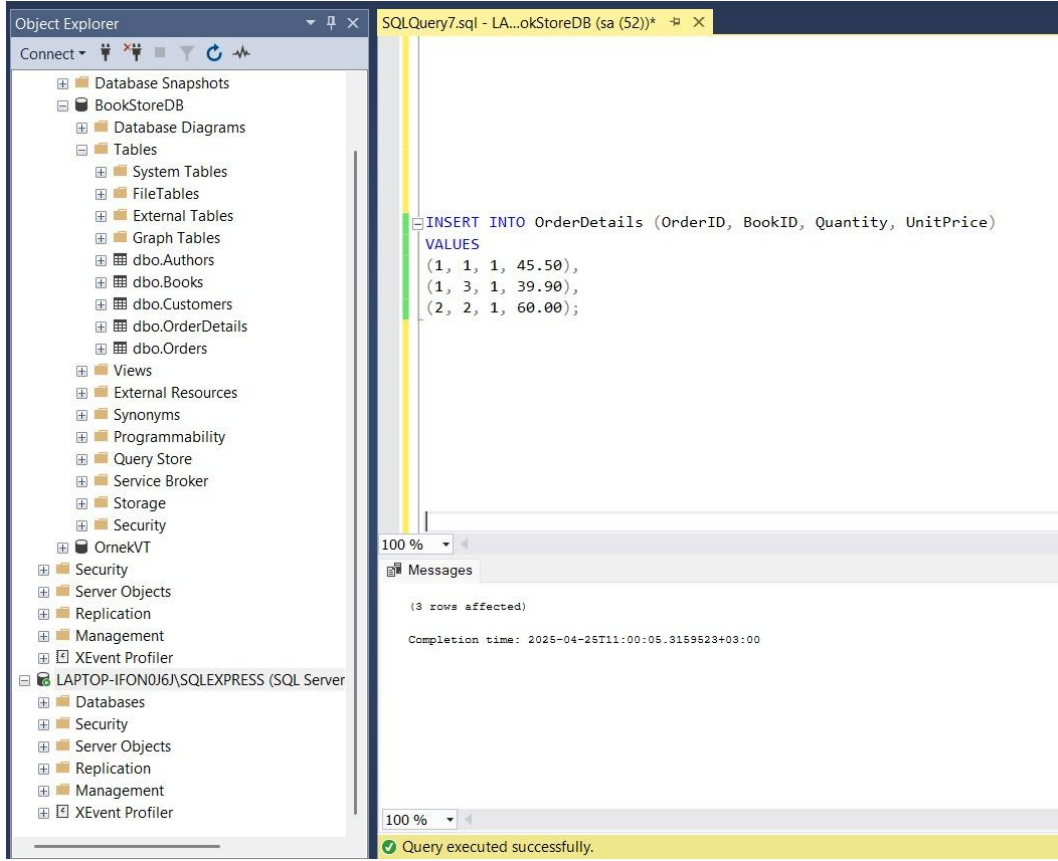
Şekil 6 Yazarlar Tablosuna Veri Ekleme



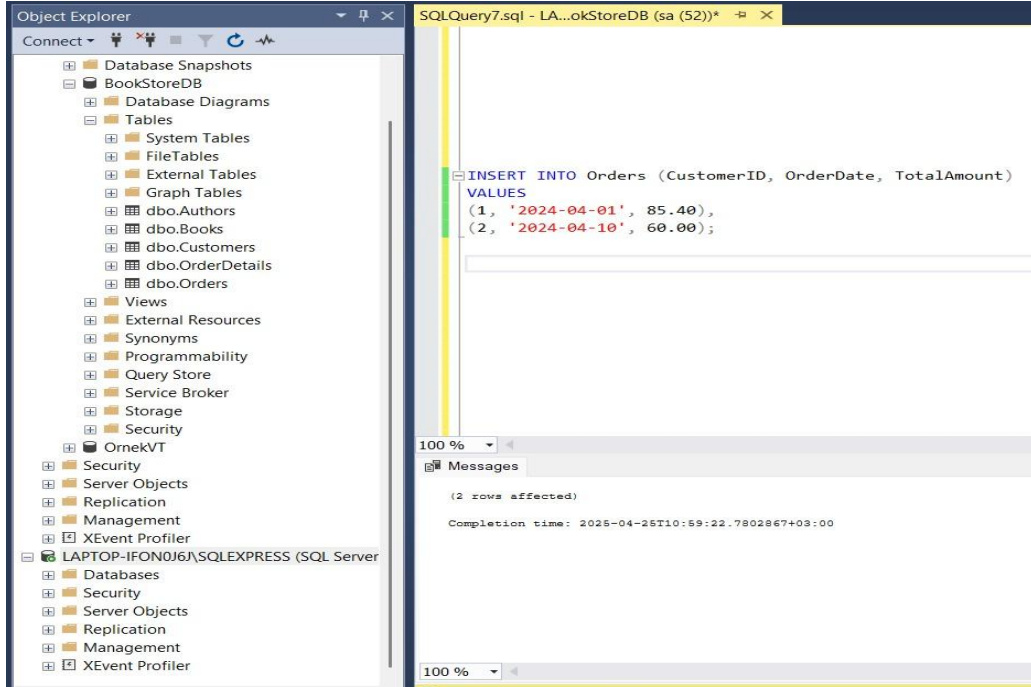
Şekil 7 Kitaplar Tablosuna veri ekleme



Şekil 8 Müşteriler Tablosuna veri ekleme



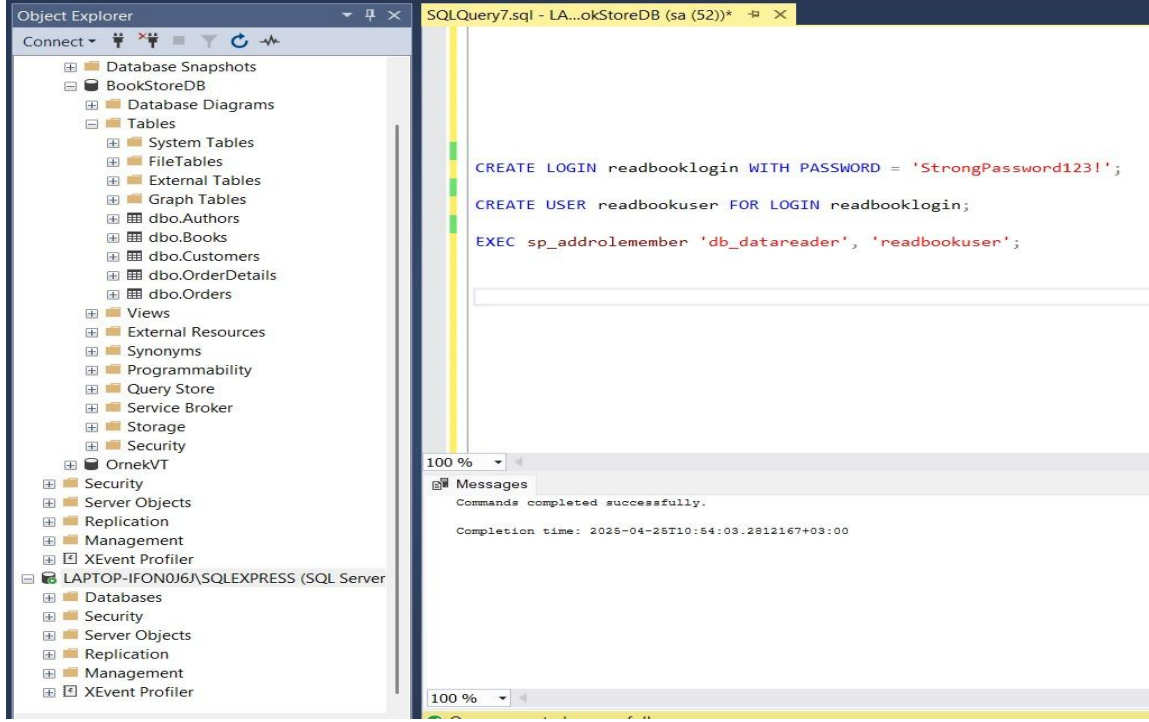
Şekil 9 Sipariş Detay Tablosuna veri ekleme



Şekil 10 Siparişler Tablosuna veri ekleme

4. Okuma Yetkisine Sahip Login Oluşturulması

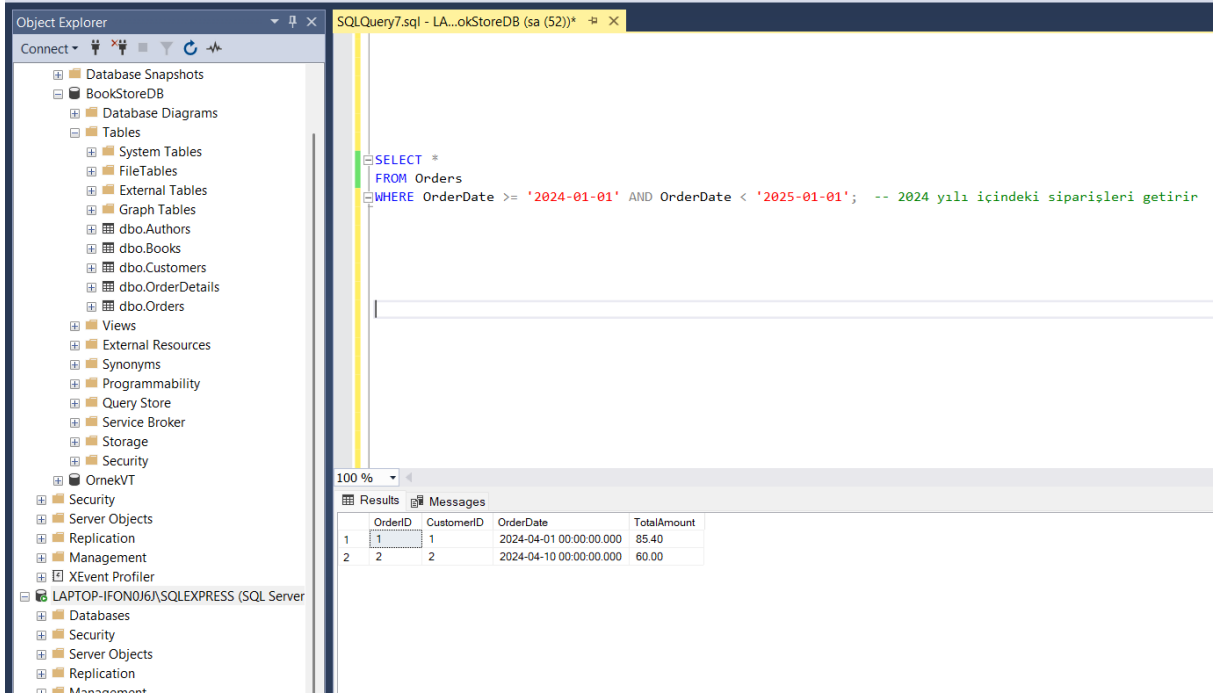
Bu SQL komutları dizisi, bir veritabanında yalnızca okuma yetkisi bulunan bir kullanıcı oluşturmak için kullanılır. Bu işlem genellikle sadece raporlama veya veri görüntüleme amacıyla kullanılan kullanıcılar için uygulanır.



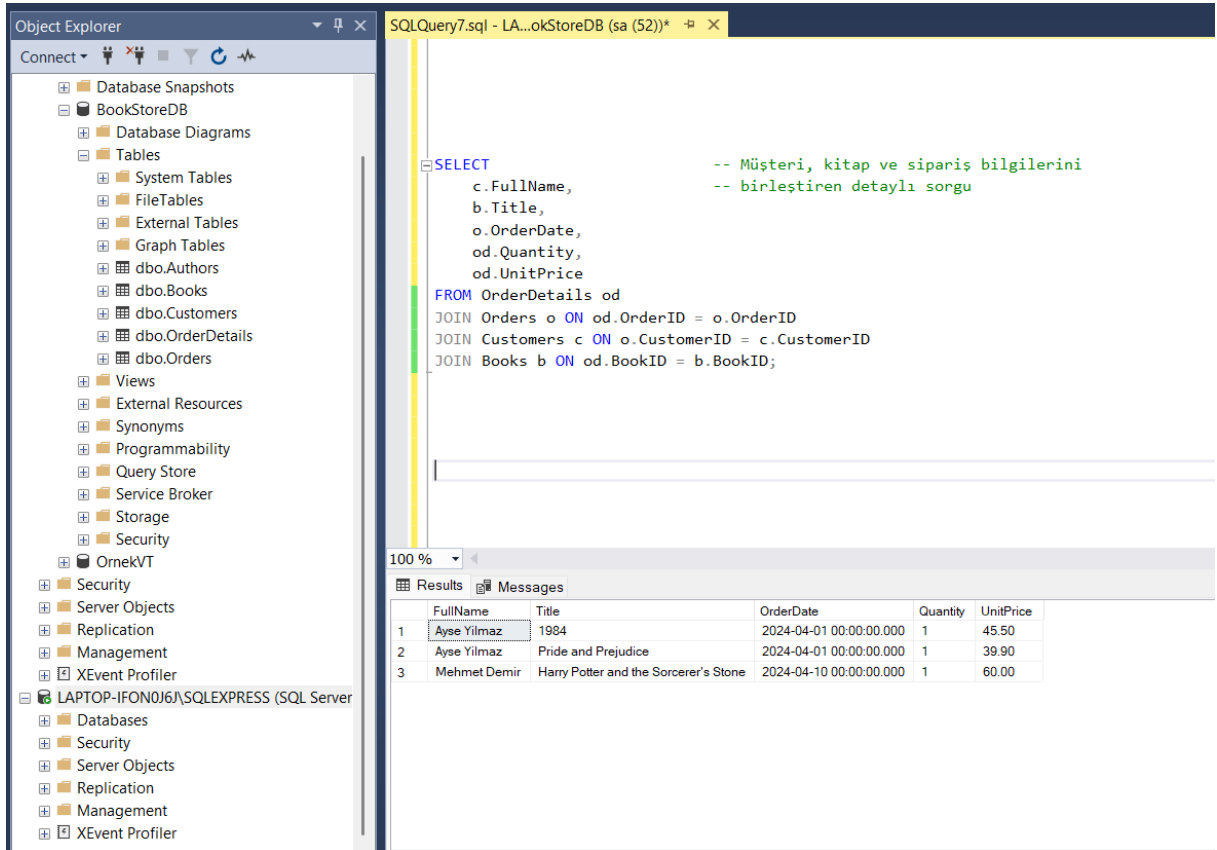
Şekil 11 Login Oluşturma

5. Temel Sorgular

- OrderDate >= '2024-01-01': 1 Ocak 2024 ve sonrasındaki siparişleri dahil eder.
- OrderDate < '2025-01-01': 1 Ocak 2025 tarihinden önceki siparişleri dahil eder.
- Bu iki koşul bir araya geldiğinde, sorgu **2024 yılına ait (1 Ocak- 31 Aralık) tüm siparişleri** getirir.
- Bu yöntem, saat bilgisi içeren datetime sütunlarında da güvenilir sonuç verir; çünkü '2025-01-01' tarihinden küçük olan tüm kayıtlar alınır, yani 2024 yılının son saniyesine kadar olanlar dahil edilir.

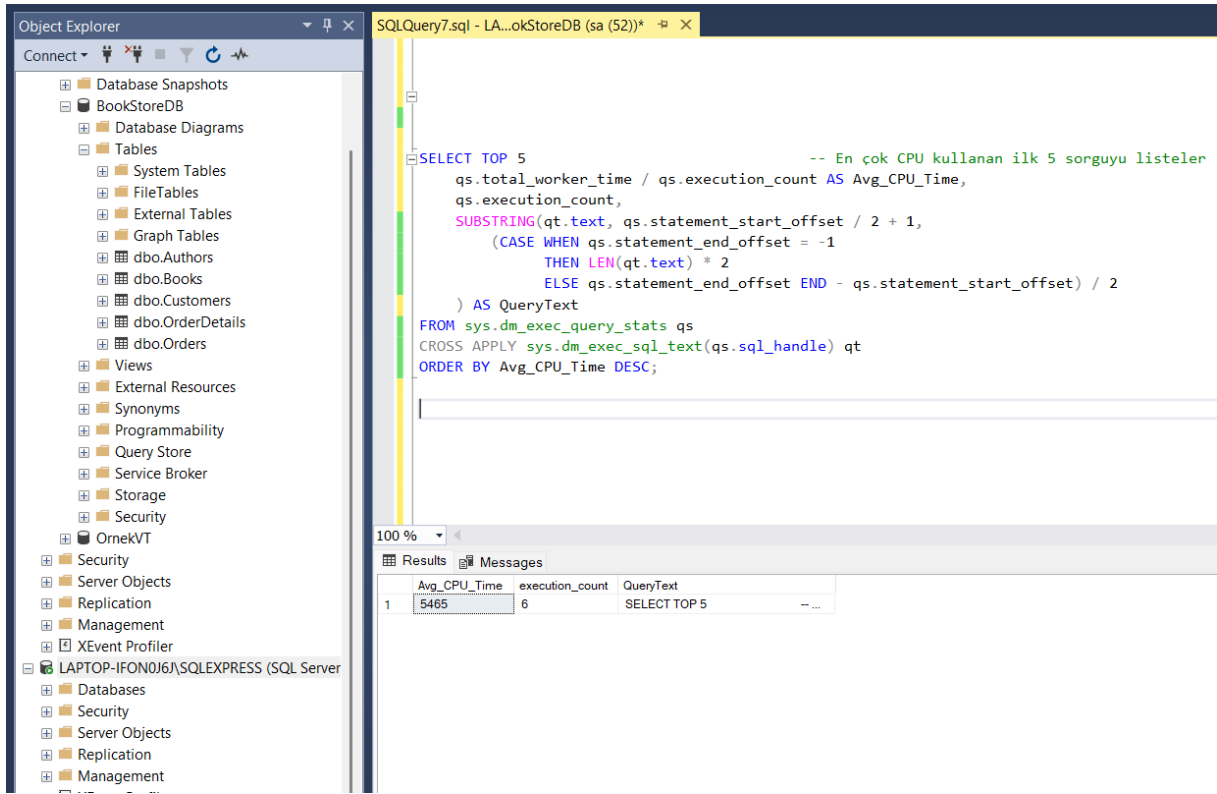


Şekil 12 2024 Yılı İçindeki Siparişler



Şekil 13 Müşteri, Kitap ve Sipariş Bilgilerini Birleştiren Sorgu

6. Performans ve İndeks Analizleri

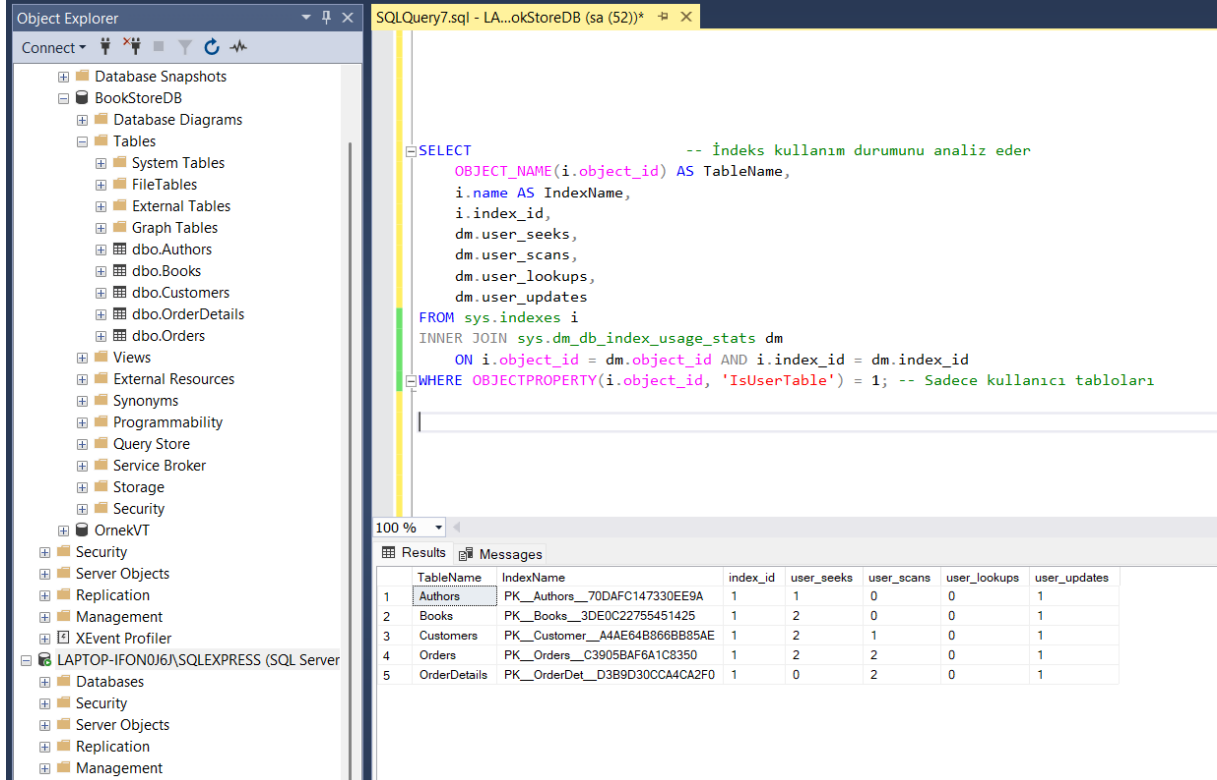


The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure of 'BookStoreDB'. The main window shows a SQL query in 'SQLQuery7.sql' that selects the top 5 queries based on average CPU time. The query uses the 'sys.dm_exec_query_stats' and 'sys.dm_exec_sql_text' system views. The results pane at the bottom shows the execution of the query, displaying the average CPU time, execution count, and the query text for the top 5 queries.

```
SELECT TOP 5 -- En çok CPU kullanan ilk 5 sorguyu listeler
    qs.total_worker_time / qs.execution_count AS Avg_CPU_Time,
    qs.execution_count,
    SUBSTRING(qt.text, qs.statement_start_offset / 2 + 1,
        (CASE WHEN qs.statement_end_offset = -1
            THEN LEN(qt.text) * 2
            ELSE qs.statement_end_offset END - qs.statement_start_offset) / 2
        ) AS QueryText
FROM sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt
ORDER BY Avg_CPU_Time DESC;
```

	Avg_CPU_Time	execution_count	QueryText
1	5465	6	SELECT TOP 5

Şekil 14 En Çok CPU Kullanan İlk 5 Sorgu Listeleme



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure of 'BookStoreDB'. The main window shows a SQL query in 'SQLQuery7.sql' that analyzes index usage. The query joins the 'sys.indexes' and 'sys.dm_db_index_usage_stats' system views to retrieve information about index usage, including the number of seeks, scans, lookups, and updates. The results pane at the bottom shows the execution of the query, displaying the table name, index name, index ID, and the number of seeks, scans, lookups, and updates for each index.

```
SELECT -- İndeks kullanım durumunu analiz eder
    OBJECT_NAME(i.object_id) AS TableName,
    i.name AS IndexName,
    i.index_id,
    dm.user_seeks,
    dm.user_scans,
    dm.user_lookups,
    dm.user_updates
FROM sys.indexes i
INNER JOIN sys.dm_db_index_usage_stats dm
    ON i.object_id = dm.object_id AND i.index_id = dm.index_id
WHERE OBJECTPROPERTY(i.object_id, 'IsUserTable') = 1; -- Sadece kullanıcı tabloları
```

	TableName	IndexName	index_id	user_seeks	user_scans	user_lookups	user_updates
1	Authors	PK_Authors__70DAFC147330EE9A	1	1	0	0	1
2	Books	PK_Books__3DE0C22755451425	1	2	0	0	1
3	Customers	PK_Customer__A4AE64B866BB85AE	1	2	1	0	1
4	Orders	PK_Orders__C3905BAF6A1C8350	1	2	2	0	1
5	OrderDetails	PK_OrderDet__D3B9D30CCA4CA2F0	1	0	2	0	1

Şekil 15 İndeks Kullanım Durumunu Analiz Etme

7. Şema Değişikliklerini İzleme

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'BookStoreDB'. The main pane shows a SQL query window with the following code:

```
CREATE TABLE SchemaChangeLog (
    LogID INT PRIMARY KEY IDENTITY(1,1),
    EventData XML,
    ChangeDate DATETIME DEFAULT GETDATE()
);
GO
```

Below the query window, the Results pane shows the execution results of the CREATE TABLE statement. The table has columns: LogID, EventData, and ChangeDate. The first row shows LogID 1, EventData <EVENT_INSTANCE><EventType>CREATE_TABLE<EventTy..., and ChangeDate 2025-04-25 11:07:21.480.

LogID	EventData	ChangeDate
1	<EVENT_INSTANCE><EventType>CREATE_TABLE<EventTy...	2025-04-25 11:07:21.480

The Messages pane shows the current database as 'BookStoreDB'.

LogID	ChangeDate	EventType	ObjectName	ExecutedCommand
1	2025-04-25 11:07:21.480	CREATE_TABLE	SchemaChangeLog	CREATE TABLE SchemaChangeLog (-- Şema değ...

Şekil 16 Şema Değişikliklerini Kaydedecek Tablo Oluşturma

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'BookStoreDB'. The main pane shows a SQL query window with the following code:

```
-- DDL trigger-> Veritabanı düzeyinde yapılan tüm şema değişikliklerini loglar
CREATE TRIGGER trgSchemaChange
ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS
AS
BEGIN
    INSERT INTO SchemaChangeLog (EventData)
    VALUES (EVENTDATA()); -- Olay verisini tabloya ekler
END;

SELECT * FROM sys.triggers WHERE name = 'trgSchemaChange';
```

Below the query window, the Results pane shows the execution results of the CREATE TRIGGER statement. The table has columns: name, object_id, parent_class, parent_class_desc, parent_id, type, type_desc, create_date, modify_date, is_ms_shipped, is_disabled, is_not_for_replication, is_instead_of_trigger.

name	object_id	parent_class	parent_class_desc	parent_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_disabled	is_not_for_replication	is_instead_of_trigger
------	-----------	--------------	-------------------	-----------	------	-----------	-------------	-------------	---------------	-------------	------------------------	-----------------------

Şekil 17 DDL Trigger

8. Örnek Şema Değişikliği

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the 'BookStoreDB' database structure, including tables, views, and other objects. The 'Books' table is selected. In the center, the 'SQLQuery7.sql' window shows the following SQL code:

```
-- Books tablosuna yeni bir sütun ekleniyor
ALTER TABLE Books ADD Publisher1 NVARCHAR(100);

EXEC sp_help 'Books';
```

On the right, the 'Results' pane displays the output of the 'sp_help' command, showing the table structure of 'Books'.

Name	Owner	Type	Created_datetime
Books	dbo	user table	2025-04-25 10:47:17.747

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
BookID	int	no	4	10	0	no	(n/a)	(n/a)	NULL
Title	nvarchar	no	200			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
Genre	nvarchar	no	100			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
Price	decimal	no	9	10	2	yes	(n/a)	(n/a)	NULL
AuthorID	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
Publisher	nvarchar	no	200			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS

Identity	Seed	Increment	Not For Replication
BookID	1	1	0

index_name	index_description	index_keys
PK_Books__3DE0C22755451425	clustered, unique, primary key located on PRIMARY	BookID

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
FOREIGN KEY	FK_Books_AuthorID_4BAC3F29	No Action	No Action	Enabled	Is_For_Replication	AuthorID REFERENCES BookStoreDB.dbo.Authors (AuthorID)
PRIMARY KE...	PK_Books__3DE0C22755451425	(n/a)	(n/a)	(n/a)	(n/a)	BookID

Table is referenced by foreign key
BookStoreDB.dbo.OrderDetails: FK_OrderData__Boo...

Şekil 18 Kitap Tablosuna Yeni Sütun Ekleme

9. Log ve Versiyon Kontrolleri

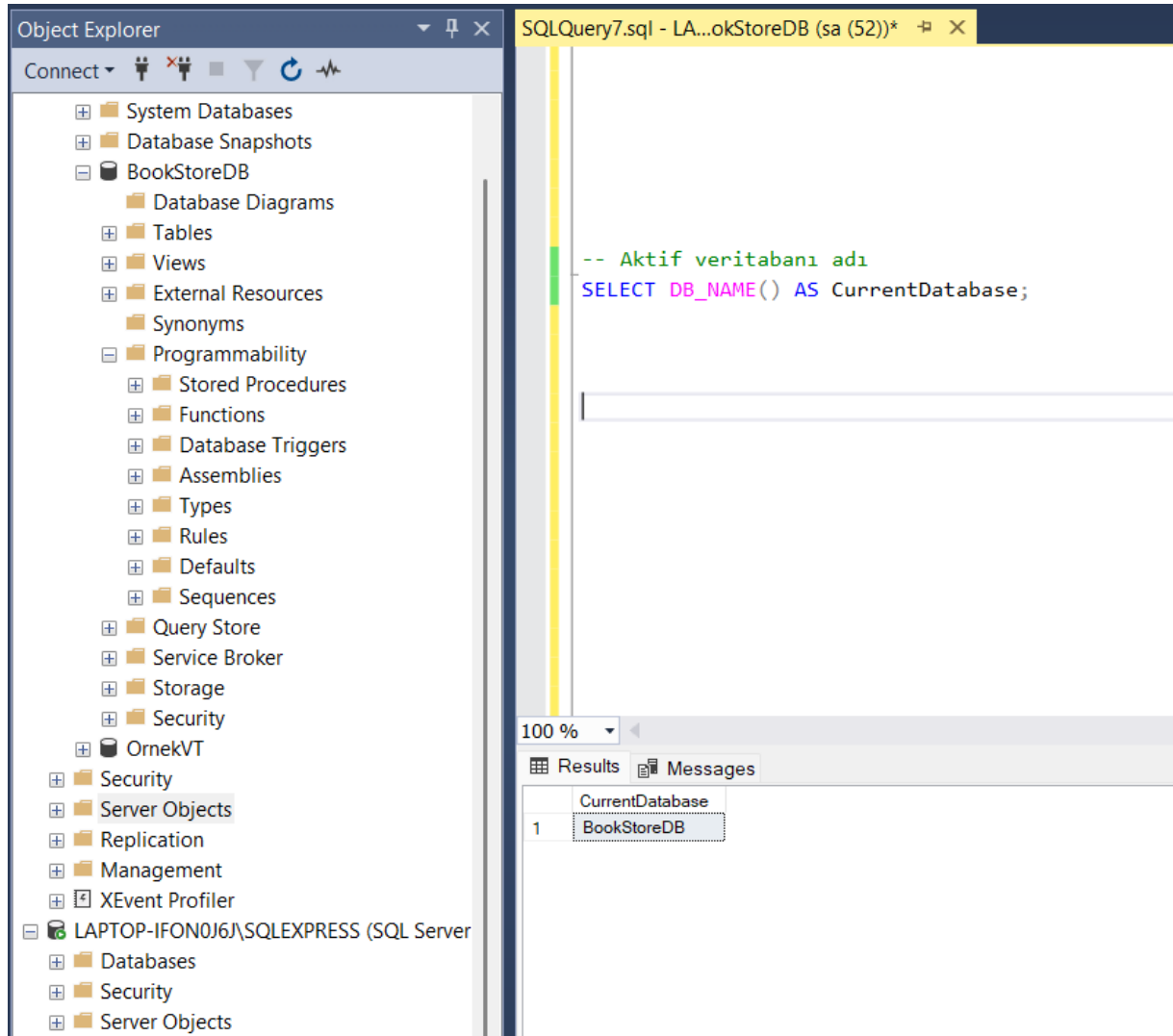
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the 'BookStoreDB' database structure. In the center, the 'SQLQuery7.sql' window shows the following SQL code:

```
-- Şema değişiklik kayıtlarını listeleme
SELECT * FROM dbo.SchemaChangeLog; -- Ham kayıtları getirir
```

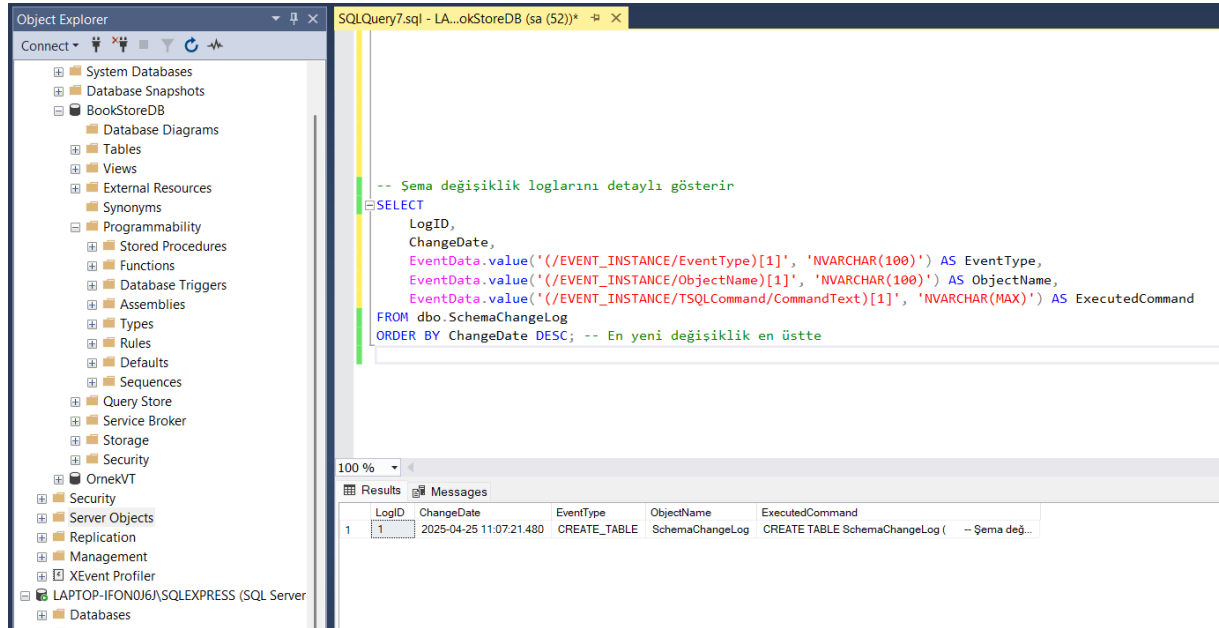
On the right, the 'Results' pane displays the output of the query, showing the schema change logs.

LogID	EventData	ChangeDate
1	<EVENT_INSTANCE><EventType>CREATE_TABLE</EventTy...	2025-04-25 11:07:21.480

Şekil 19 Şema Değişiklik Kayıtlarını Listeleme



Şekil 20 Aktif veri Tabanı



Şekil 21 Şema Değişiklik Logları

10. Kaynakça

https://www.youtube.com/watch?v=4jqJKQoWM_s

<https://github.com/JineenRihawi>