

LAB 11 (VIEWS)



Session: 2022-2026

Submitted by:

Afeera Fatima 2022-CS-151

Supervised by:

Nazeef ul Haq

Course:

CS-262L Database Systems

Department of Computer Science

University Of Engineering And Technology,

Lahore, Pakistan

Contents

1	Create SQL VIEW in SQL Serve	4
1.1	Method 1	4
1.2	Method 2	4
1.3	Method 3	5
1.4	Method 4	6
2	Retrieve Data From View in SQL Server	6
2.1	Method 1	6
2.2	Method 2	7
3	Dropping a View in SQL Server	8
3.1	Syntax	8
4	Renaming the View in SQL Server	8
4.1	Syntax	8
4.2	Example	8
5	Getting Information about a view	8
5.1	Syntax	9
5.2	Output	9
6	Alter View in SQL Server	9
6.1	Syntax	9
6.2	Example	10
7	Refreshing a View in SQL Server	10
7.1	Refreshing the View	10
8	SchemaBinding a VIEW	10
8.1	Query	11
8.2	Example	11
9	Encrypt a view in SQL Server	11
9.1	Query	12
9.2	Example	12
10	Check Option	13
10.1	Example	13
10.1.1	Without Check Option	13
10.1.2	With Check Option	14

11	DML Query In View	15
11.1	Conditions for Implementing DML Queries in a View	15
11.2	Example	15
12	System Define Views	16
12.1	Information Schema	16
12.1.1	Example	16
12.2	Catalog View	17
12.2.1	Example	18

1 Create SQL VIEW in SQL Serve

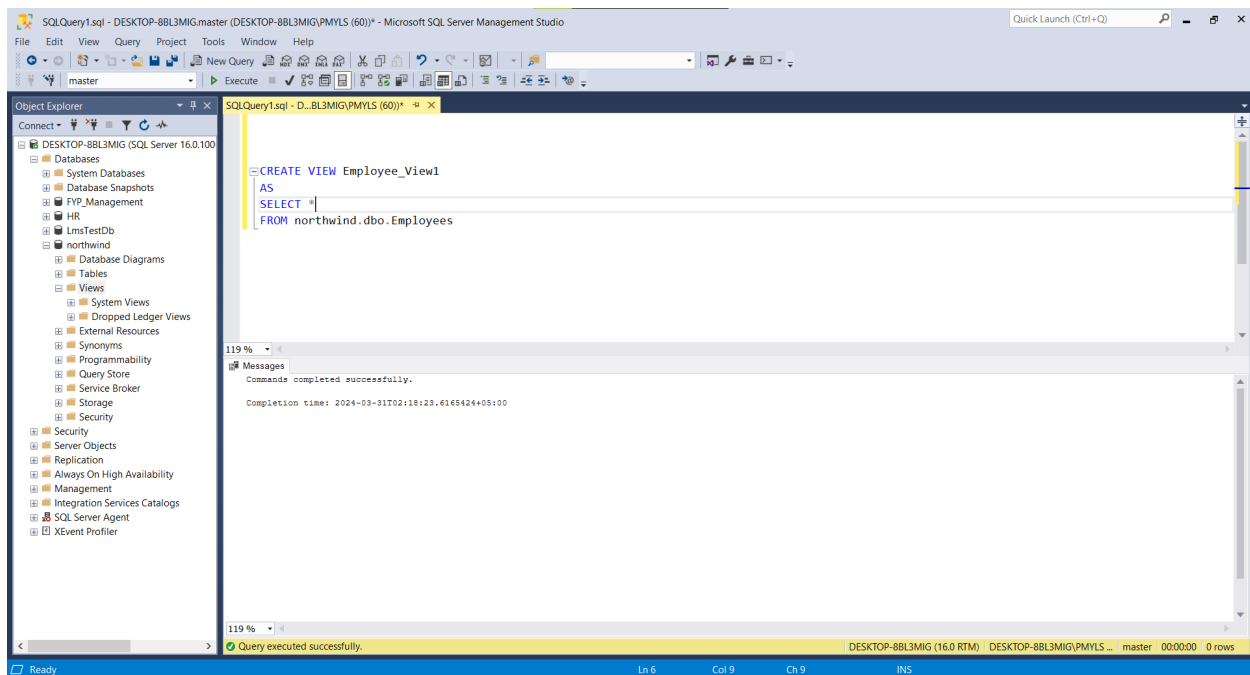
Click on your Database and expand the view item there you'll find the user stored views and system-defined views. Now open a new query create a view and execute it. It will automatically be made in the view section after refreshing views.

1.1 Method 1

We can select all columns of a table.

```
CREATE VIEW Employee_View1  
AS  
SELECT *  
FROM Employee_Details
```

The following example demonstrates that:

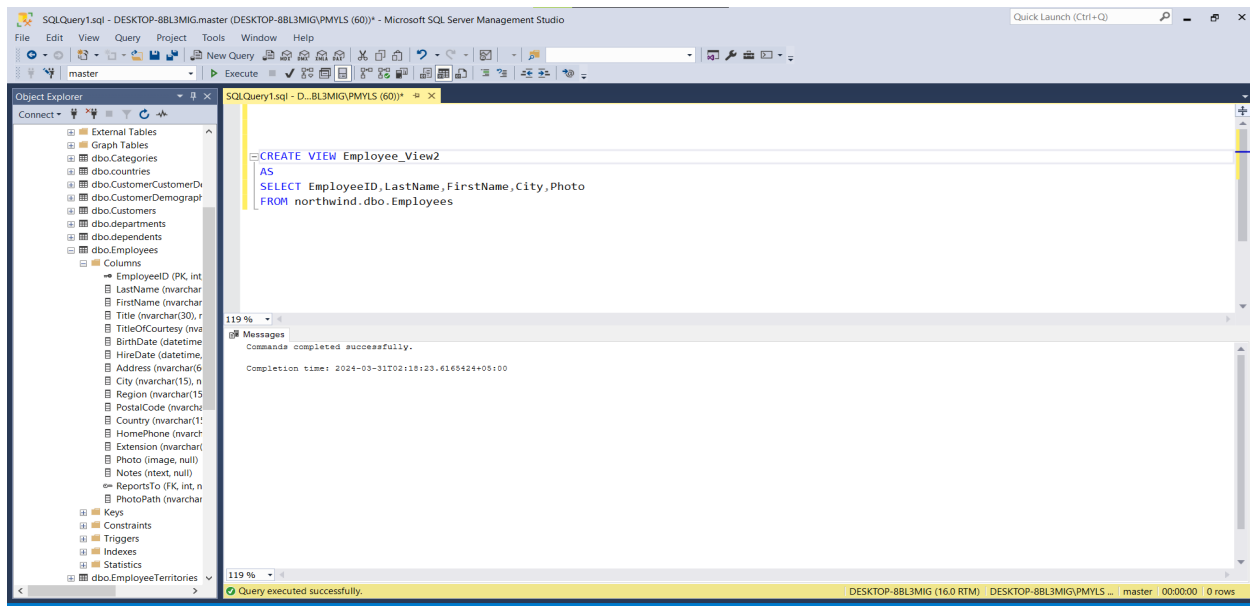


1.2 Method 2

We can select specific columns of a table.

```
CREATE VIEW Employee_View2  
AS  
SELECT Emp_Id, Emp_Name, Emp_City  
FROM Employee_Details
```

The following example demonstrates that:

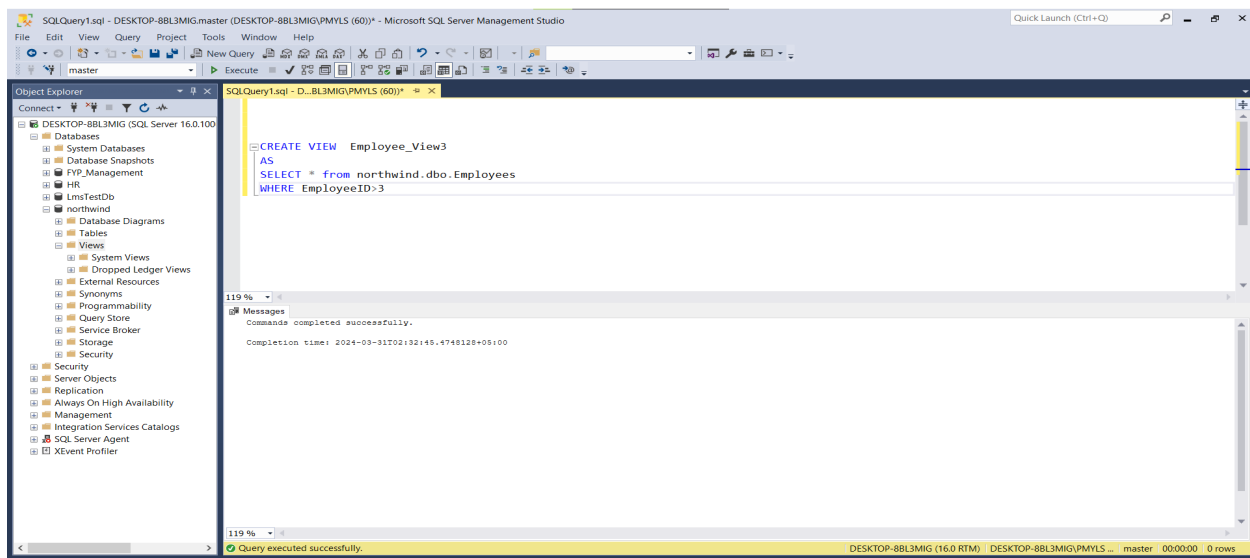


1.3 Method 3

We can select columns from a table with specific conditions.

```
CREATE VIEW Employee_View3
AS
SELECT * from Employee_Details
WHERE Emp_Id>3
```

The following example demonstrates that:



1.4 Method 4

We can create a view that will hold the columns of different tables.

CREATE VIEW Employee_View4

AS

SELECT Employee_Details.Emp_Id, Employee_Details.Emp_Name, Employee_Details.MobileNo **from** Employee_Details

LEFT OUTER JOIN

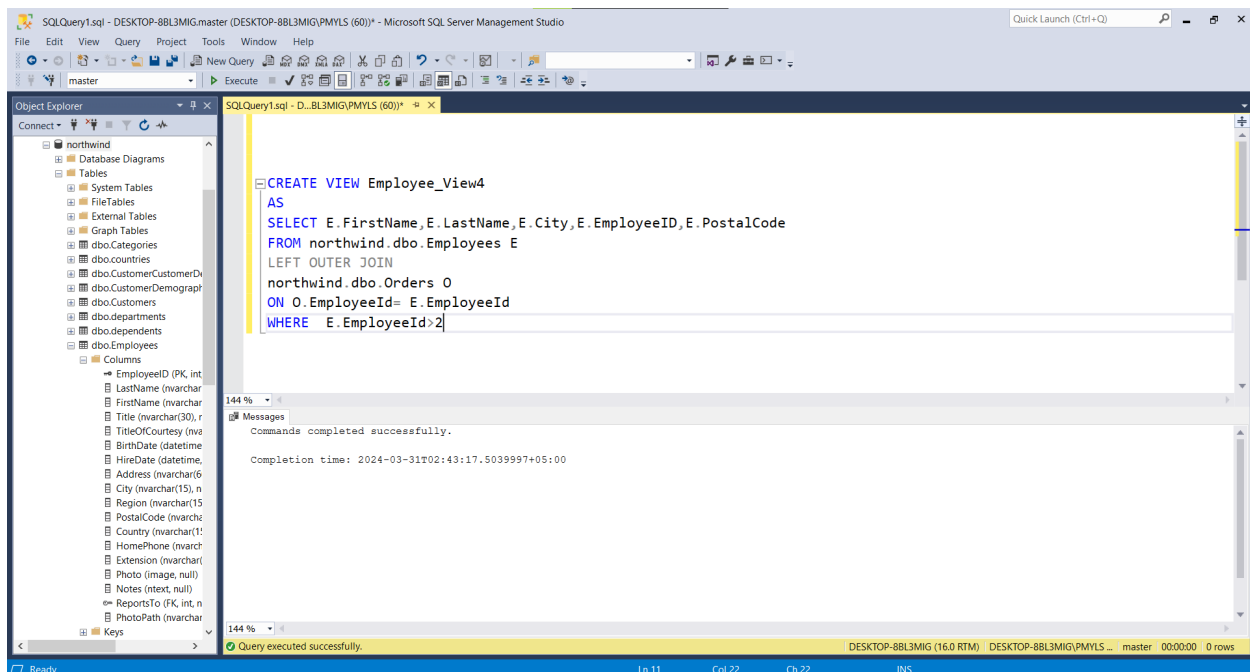
Employee_Contact

ON

Employee_Details .Emp_Id= Employee_Contact.Emp_Id

WHERE Employee_Details.Emp_Id>2

The following example demonstrates that:



2 Retrieve Data From View in SQL Server

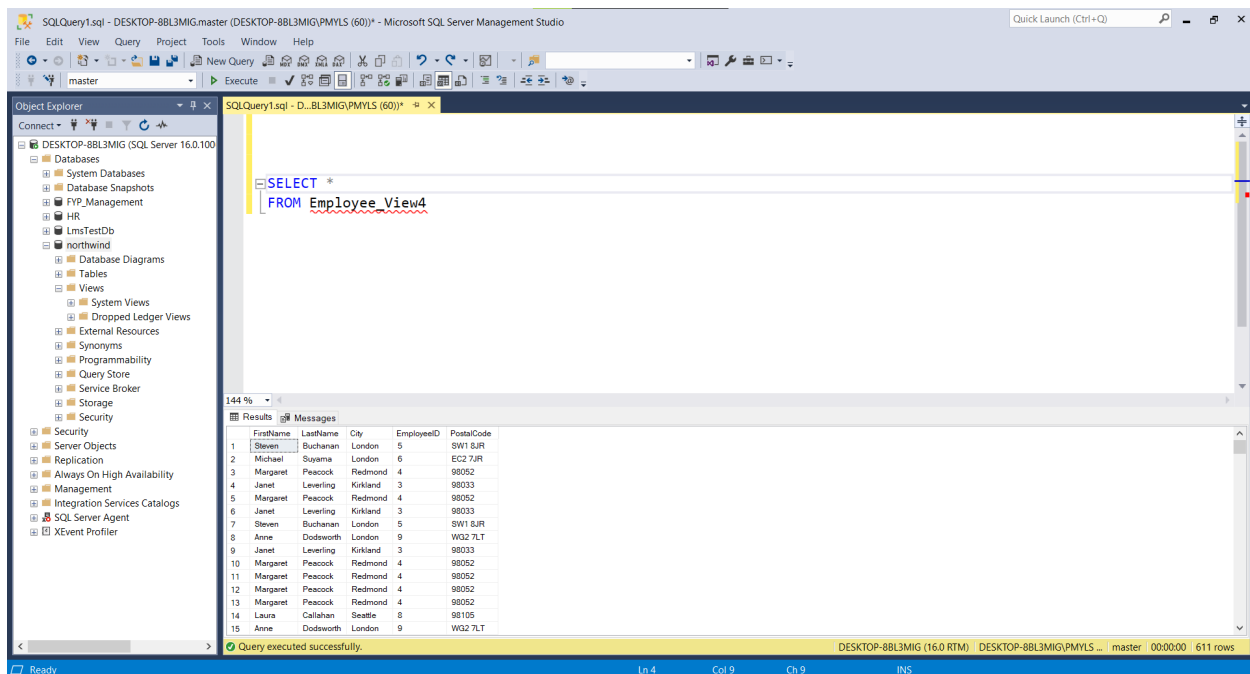
This SQL CREATE VIEW example would create a virtual table based on the result set of the select statement. Now we can retrieve data from a view as follows:

2.1 Method 1

This query shows that we can select all the columns

SELECT *

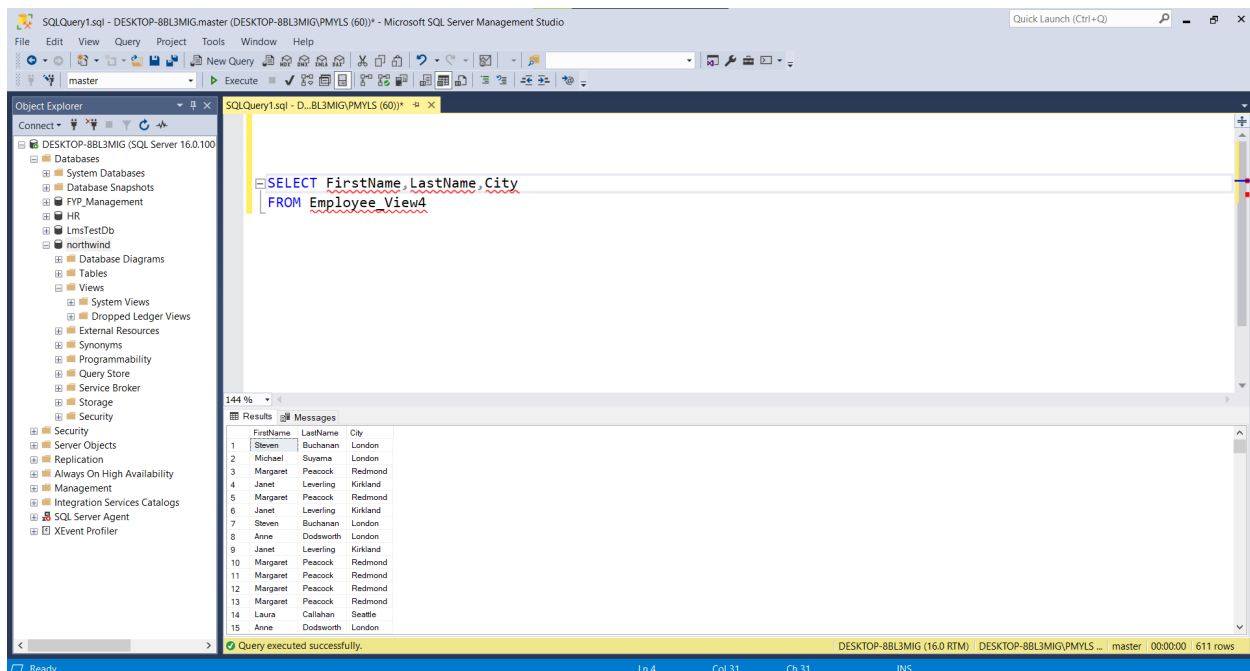
FROM Employee_View4



2.2 Method 2

This query shows that we can select all the columns

```
SELECT Emp_Id , Emp_Name , Emp_Salary  
FROM Employee_View4
```



3 Dropping a View in SQL Server

We can use the DROP command to drop a view. For example, to drop the view `Employee_View1`, we can use the following statement:

3.1 Syntax

```
DROP VIEW Employee_View1 ;
```

4 Renaming the View in SQL Server

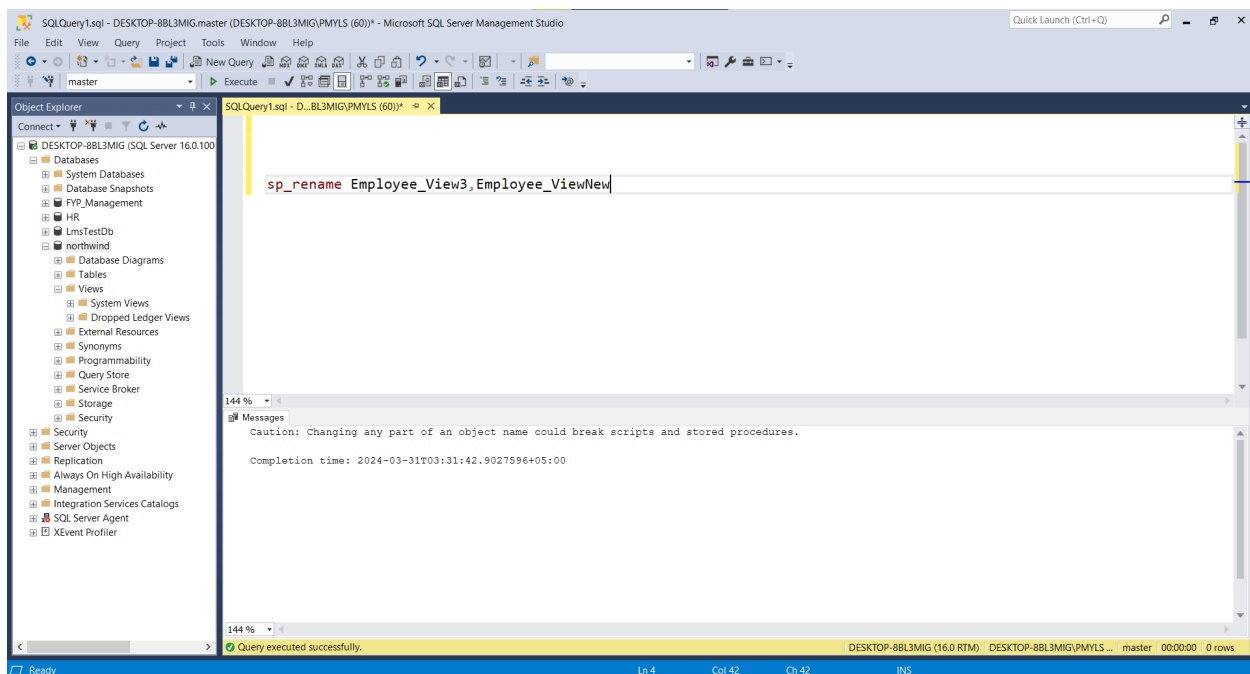
We can use the `sp_rename` system procedure to rename a view. The syntax of the `sp_rename` command is given below:

4.1 Syntax

```
sp_rename OldViewName , NewViewName ;
```

4.2 Example

```
sp_rename Employee_View3 , Employee_ViewNew ;
```



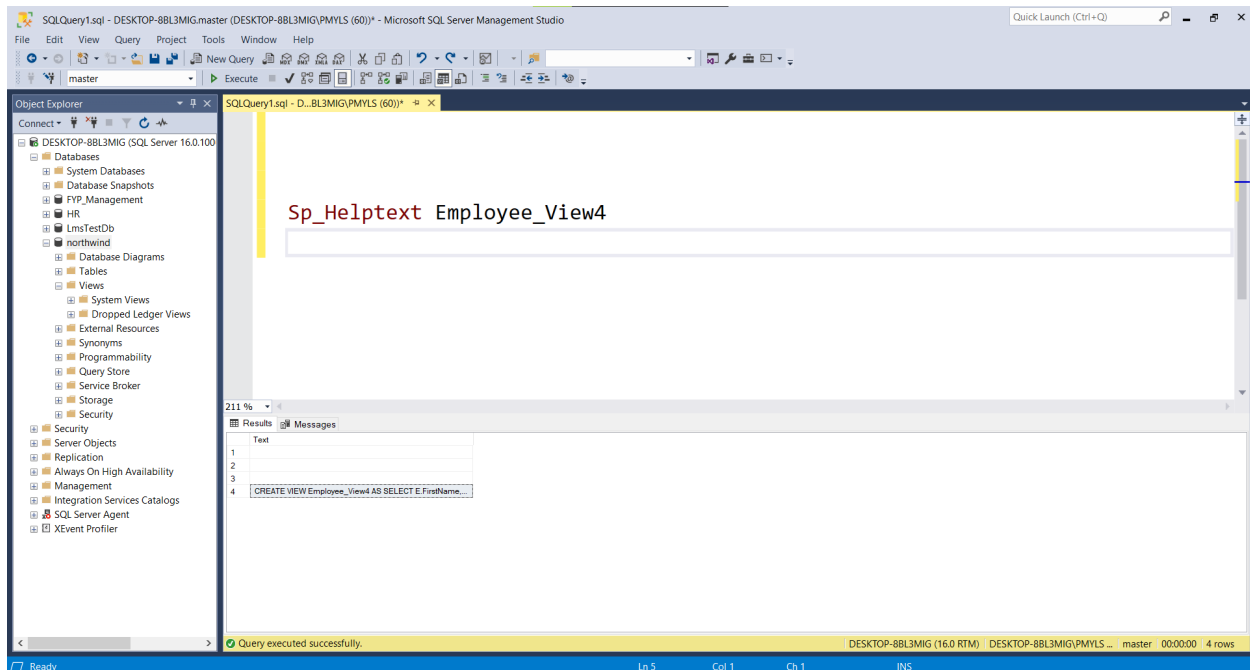
5 Getting Information about a view

We can retrieve all the information of a view using the `Sp_Helptext` system Stored Procedure.

5.1 Syntax

```
Sp_Helptext View_Name
```

5.2 Output



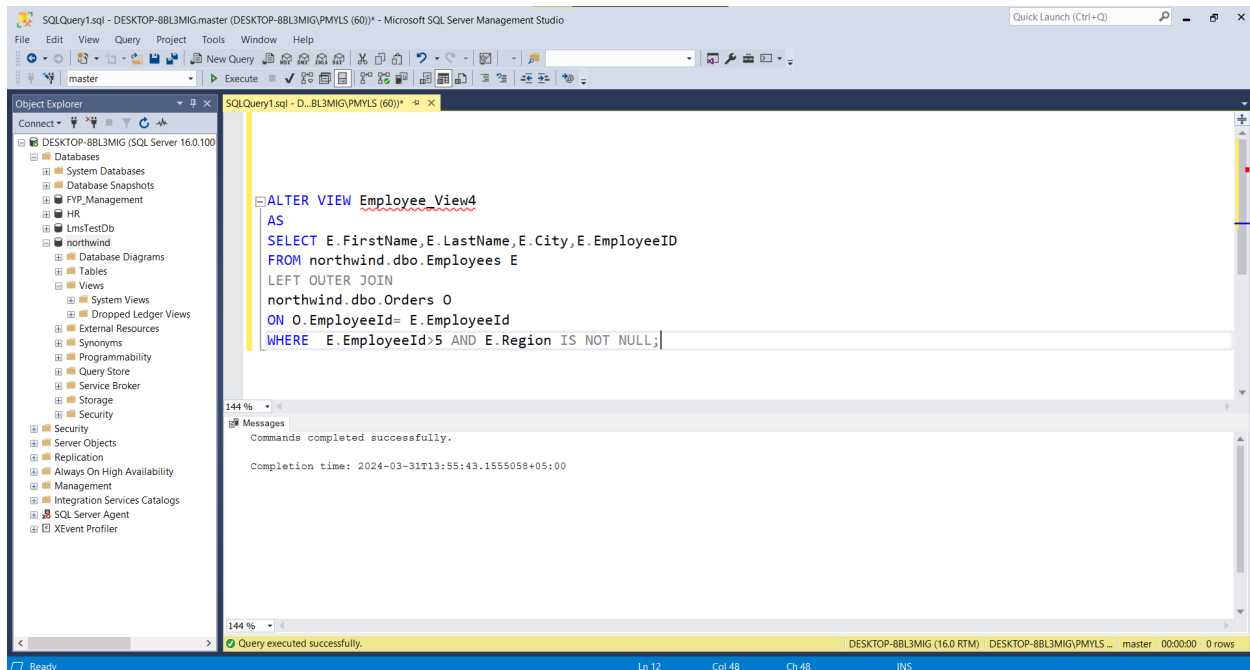
6 Alter View in SQL Server

We can alter the schema or structure of a view. In other words, we can add or remove some columns or change some conditions that are applied in a predefined view.

6.1 Syntax

```
Alter View Employee_View4
AS
SELECT Employee_Details.Emp_Id, Employee_Details.Emp_Name,
Employee_Details.Emp_Salary, Employee_Contact.MobileNo
FROM Employee_Details
LEFT OUTER JOIN
Employee_Contact
ON
Employee_Details.Emp_Id= Employee_Contact.Emp_Id
WHERE Employee_Details.Emp_Id>5
AND Employee_Details.Emp_City= 'Alwar '
```

6.2 Example



7 Refreshing a View in SQL Server

When a view is created, it represents a snapshot of the underlying tables' data at the time of its creation. If the original table is modified after the view is created, these changes are not automatically reflected in the view. Therefore, to synchronize the view with the latest data from the underlying table, we need to refresh it.

7.1 Refreshing the View

To refresh the view `Employee_View1`, you can use the following SQL command:

```
EXEC sp_refreshview Employee_View1;
```

This command refreshes the metadata for the specified view, ensuring that it reflects the latest schema changes in the underlying tables.

8 SchemaBinding a VIEW

We saw that if we add a new column into the table then we must refresh the view.

Such a way if we change the data type of any column in a table then we should refresh the view. If we want to prevent any type of change in a base table then we can use the concept

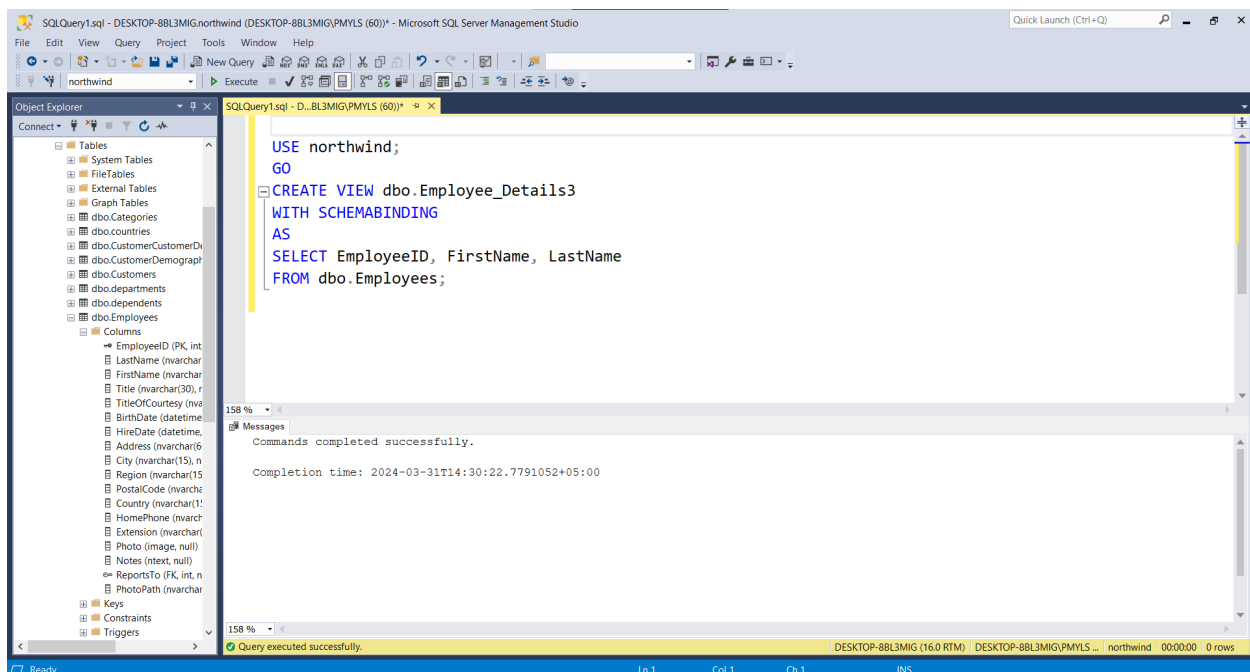
of SCHEMABINDING. It will lock the tables being referred to by the view and restrict all kinds of changes that may change the table schema (no Alter command).

We can't specify "Select * from tablename" with the query. We need to specify all the column names for reference

8.1 Query

```
CREATE VIEW Employee_Details3  
WITH SCHEMABINDING  
AS  
SELECT Emp_Id,Emp_Name,Emp_Salary ,Emp_City  
FROM DBO.Employee_Details
```

8.2 Example



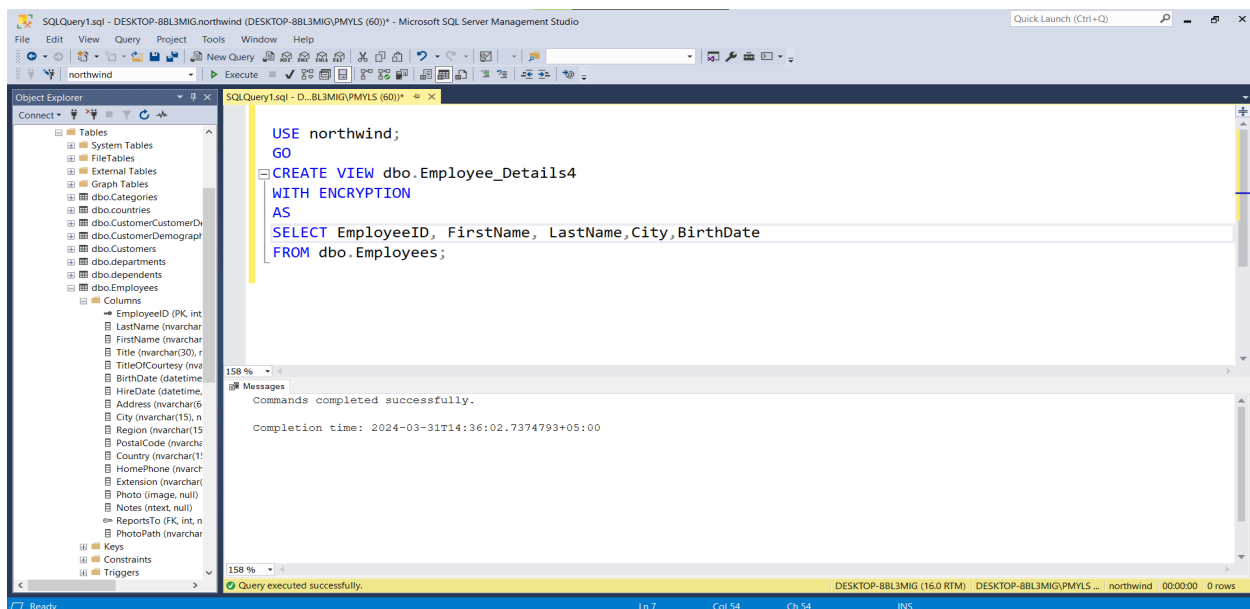
9 Encrypt a view in SQL Server

The "WITH ENCRYPTION" option can encrypt any views. That means it will not be visible via SP_HELPTEXT. This option encrypts the definition. This option encrypts the definition of the view. Users will not be able to see the definition of the view after it is created. This is the main advantage of the view where we can make it secure.

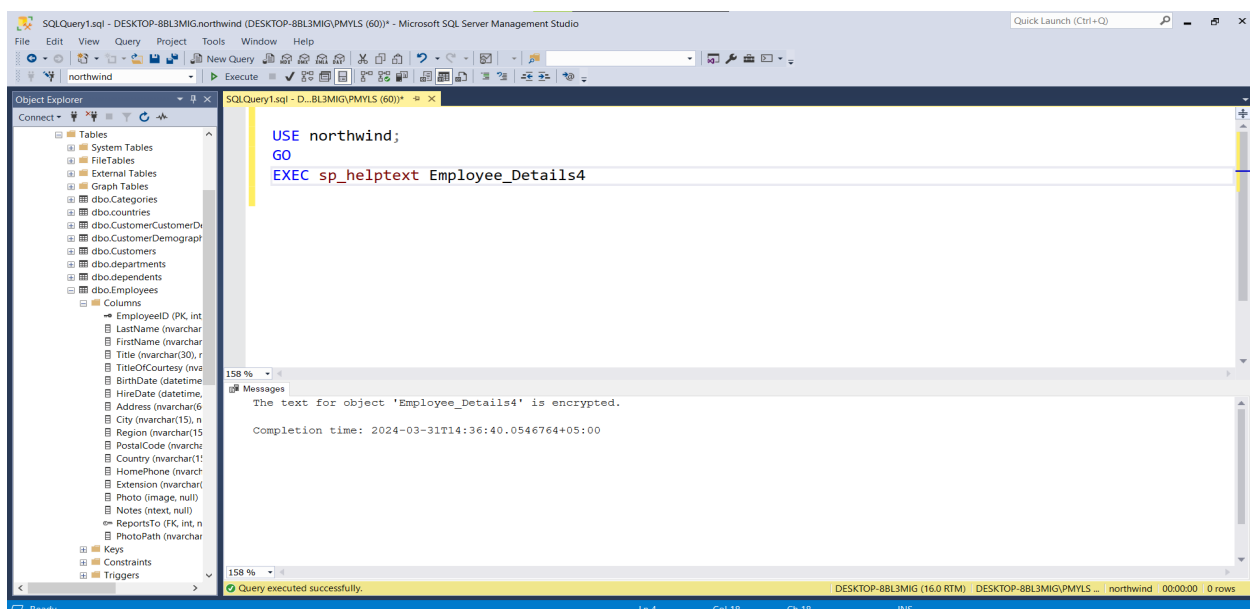
9.1 Query

```
CREATE VIEW dbo.Employee_Details4  
WITH ENCRYPTION  
AS  
SELECT EmployeeID , FirstName , LastName , City , BirthDate  
FROM dbo.Employees ;
```

9.2 Example



Now we try to retrieve the definition of the view.

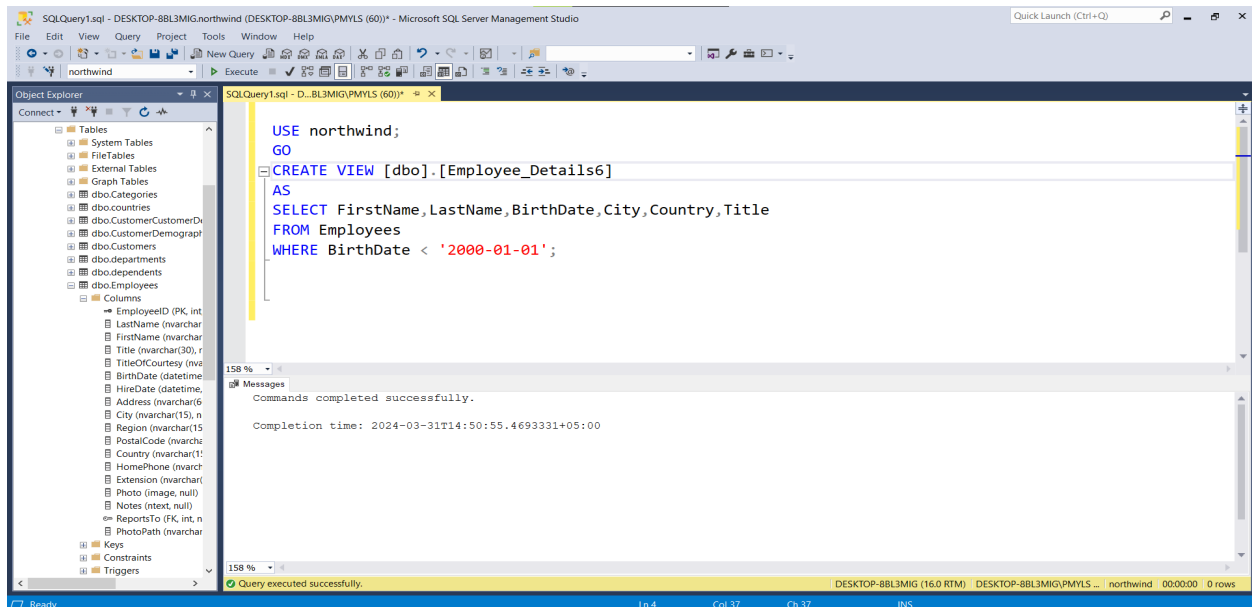


10 Check Option

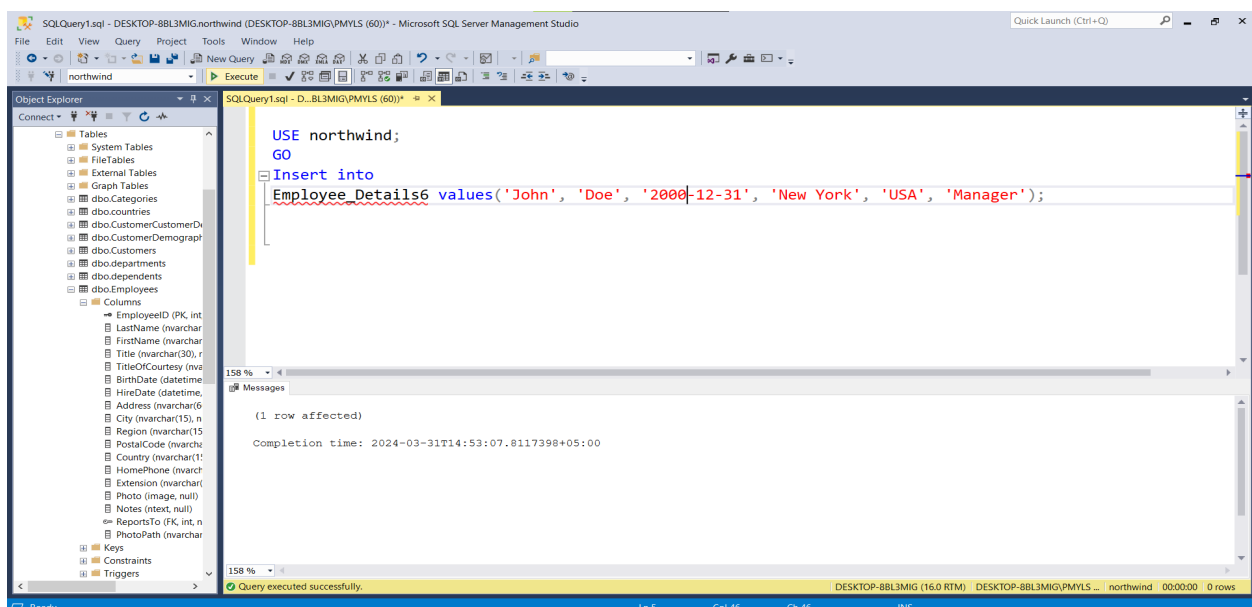
The use of the Check Option in a view is to ensure that all the Update and Insert commands must satisfy the condition in the view definition.

10.1 Example

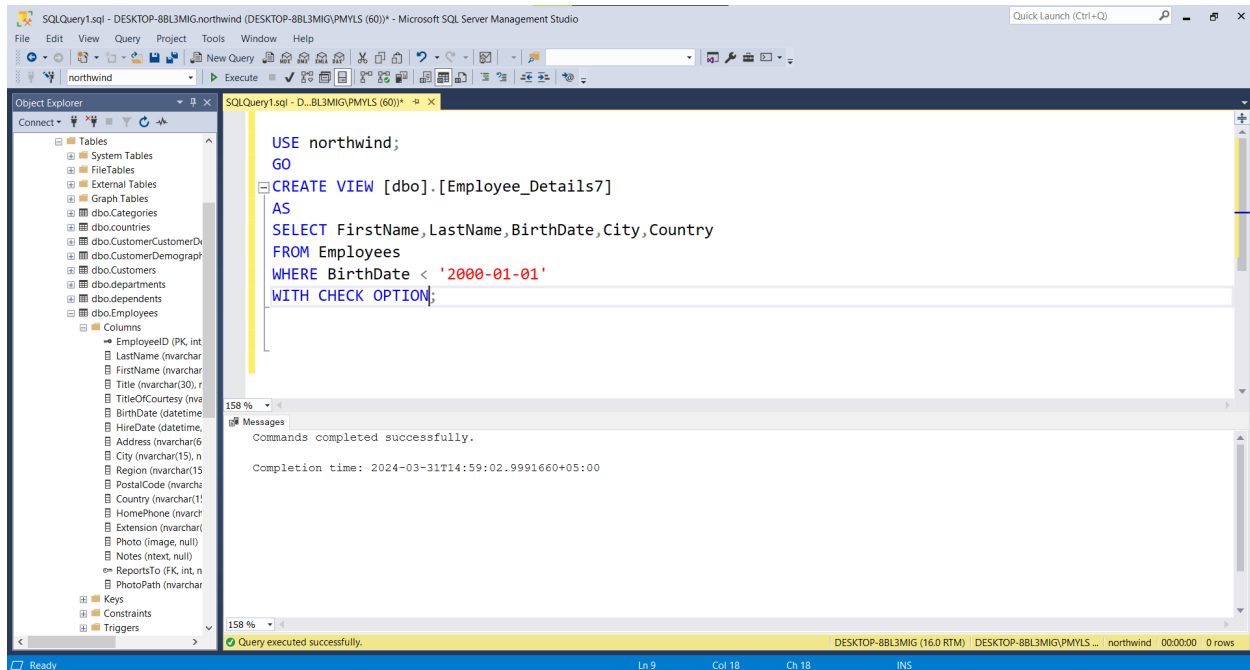
10.1.1 Without Check Option



In the preceding example, we create a view that contains all the data for which Birthdate < 2000-01-01 but we can insert the data for a Birthdate greater than 2000-01-01 as follows.

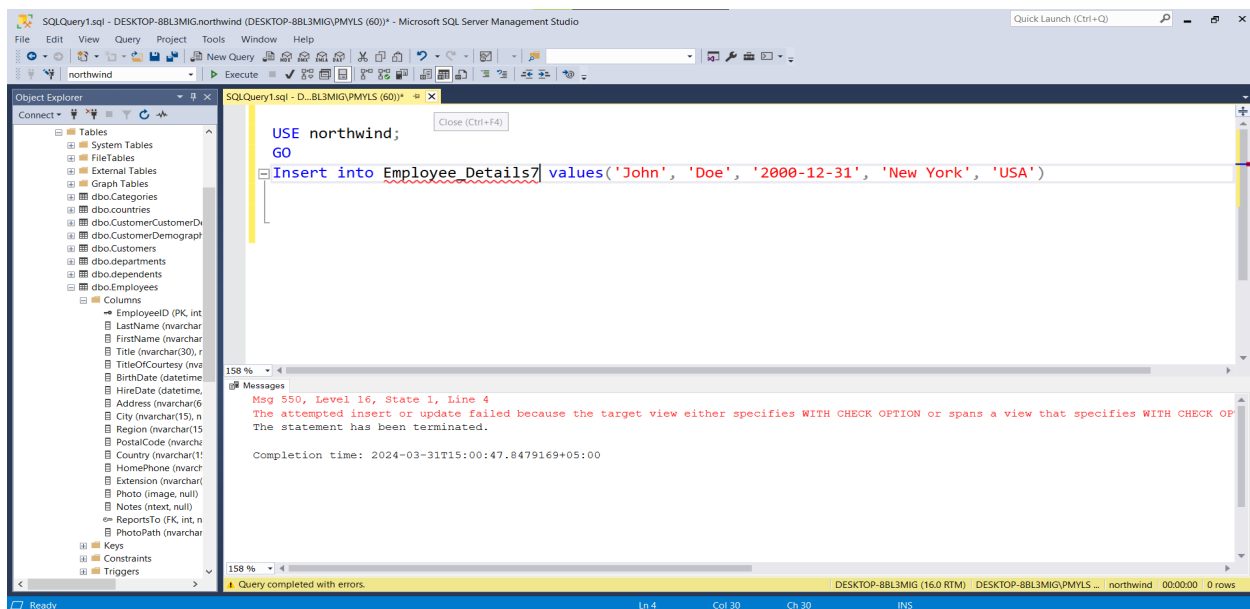


10.1.2 With Check Option



Now if we try to execute the preceding query then it will throw an error such as:

The attempted insert or update failed because the target view either specifies WITH CHECK OPTION or spans a view that specifies WITH CHECK OPTION and one or more rows resulting from the operation did not qualify under the CHECK OPTION constraint.



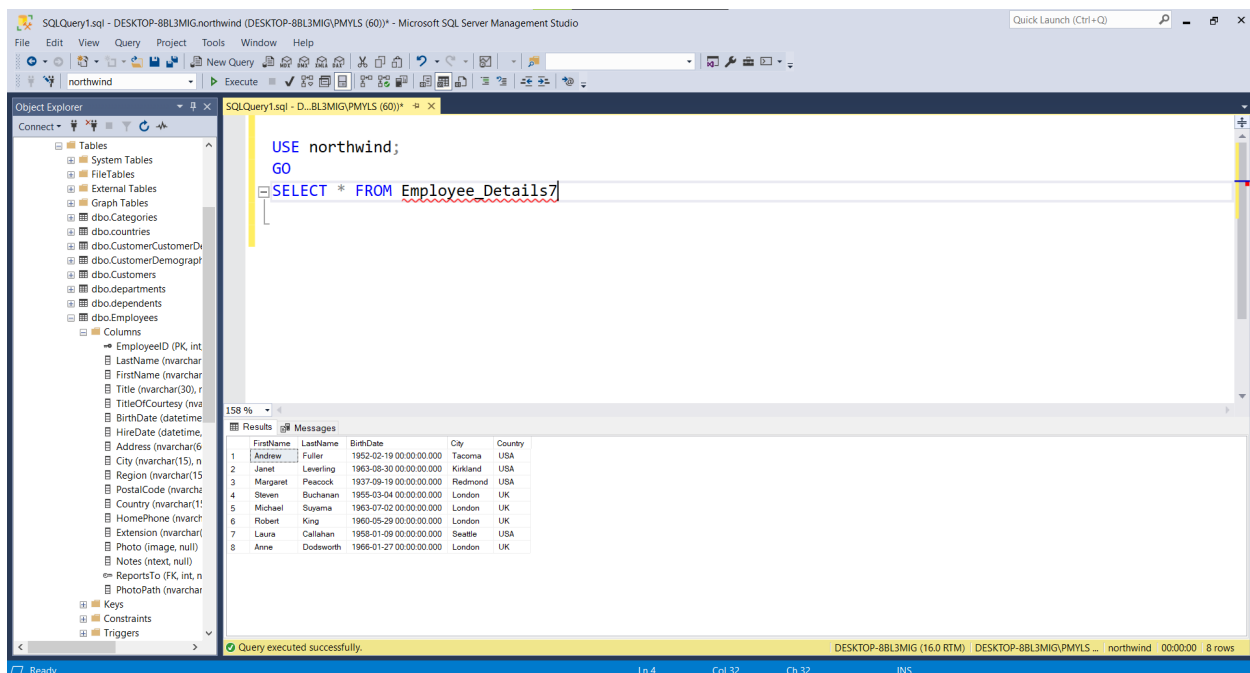
11 DML Query In View

11.1 Conditions for Implementing DML Queries in a View

For a successful implementation of a DML (Data Manipulation Language) query in a view, the following conditions should be considered:

1. The view should not contain multiple tables.
2. The view should not contain set functions.
3. The view should not use the DISTINCT keyword.
4. The view should not contain GROUP BY or HAVING clauses.
5. The view should not contain subqueries.
6. The view should not use set operators.
7. All NOT NULL columns from the base table must be included in the view in order for the INSERT query to function.

11.2 Example



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'northwind' database structure, including tables like 'Employee_Details' and 'Customers'. The central query window shows the following SQL code:

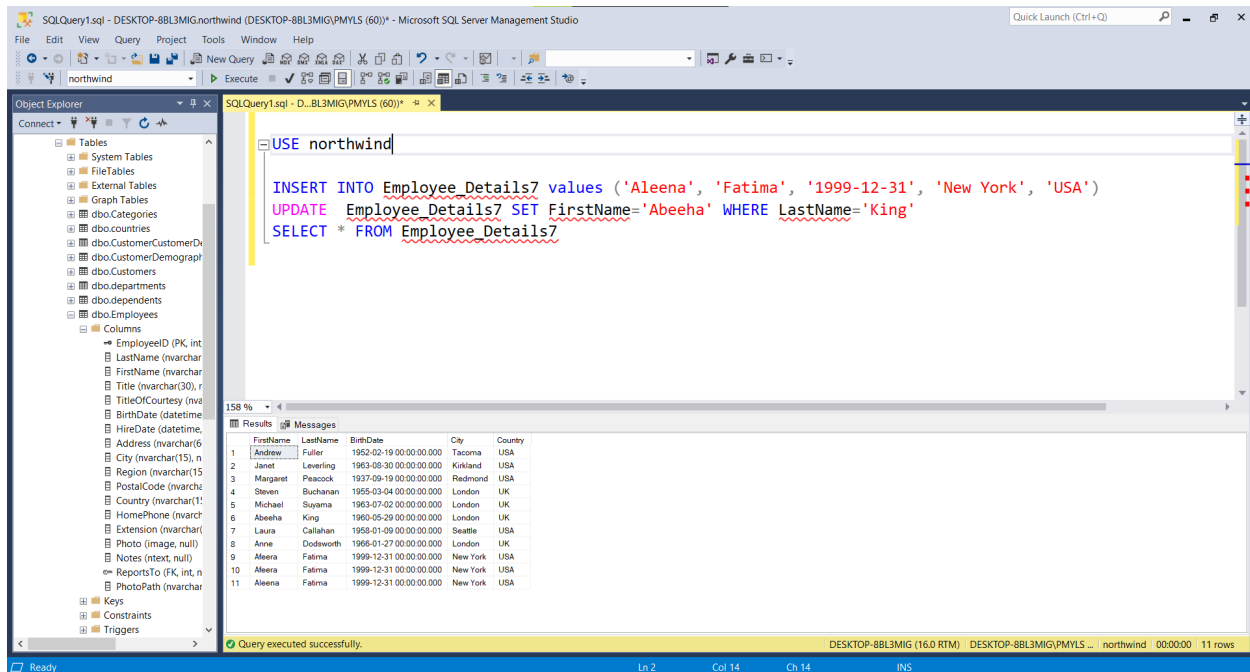
```
USE northwind;  
GO  
SELECT * FROM Employee_Details
```

The Results pane at the bottom shows the output of the query, displaying 8 rows of employee data:

FirstName	LastName	BirthDate	City	Country
Andrew	Fuller	1952-02-19 00:00:00.000	Tacoma	USA
Janet	Leverling	1963-08-30 00:00:00.000	Kirkland	USA
Margaret	Peacock	1957-09-19 00:00:00.000	Redmond	USA
Steven	Buchanan	1955-03-04 00:00:00.000	London	UK
Michael	Suyama	1963-07-02 00:00:00.000	London	UK
Robert	King	1960-05-29 00:00:00.000	London	UK
Laura	Callahan	1958-01-09 00:00:00.000	Seattle	USA
Anne	Dodsworth	1966-01-27 00:00:00.000	London	UK

The status bar at the bottom indicates 'Query executed successfully.' and '8 rows'.

In a view we can implement many types of DML query like insert, update and delete



12 System Define Views

SQL Server also contains various predefined databases like Tempdb, Master, temp. Each database has their own properties and responsibility. Master data is a template database for all other user-defined databases. A Master database contains many Predefine_View that work as templates for other databases and tables. Master databases contain nearly 230 predefined views.

These predefined views are very useful to us. Mainly we divide system views into the following two parts.

1. Information Schema
2. Catalog View

12.1 Information Schema

There are nearly 21 Information Schemas in the System. These are used for displaying the most physical information of a database, such as table and columns. An Information Schema starts from INFORMATION_SCHEMA.[View Name].

12.1.1 Example

Let us see an example: This Information Schema returns the details of all the views used by the table Employees


```
USE northwind;
GO
SELECT * FROM INFORMATION_SCHEMA.VIEW_TABLE_USAGE
WHERE TABLE_NAME='Employees'
```

VIEW_CATALOG	VIEW_SCHEMA	VIEW_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME
northwind	dbo	Employee_Details3	northwind	dbo	Employees
northwind	dbo	Employee_Details4	northwind	dbo	Employees
northwind	dbo	Employee_Details5	northwind	dbo	Employees
northwind	dbo	Employee_Details6	northwind	dbo	Employees
northwind	dbo	Employee_Details7	northwind	dbo	Employees

Query executed successfully. DESKTOP-8BL3MIG (16.0 RTM) DESKTOP-8BL3MIG\PMYLS ... northwind 00:00:00 5 rows

This Information Schema returns the information about the constraints of a table.

```
USE northwind;
GO
SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE TABLE_NAME='Employees'
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	IS_DEFERRABLE	INITIALLY_DEFERRED
northwind	dbo	PK_Employees	northwind	dbo	Employees	PRIMARY KEY	NO	NO
northwind	dbo	CK_Birthdate	northwind	dbo	Employees	CHECK	NO	NO
northwind	dbo	FK_Employees_Employees	northwind	dbo	Employees	FOREIGN KEY	NO	NO

Query executed successfully. DESKTOP-8BL3MIG (16.0 RTM) DESKTOP-8BL3MIG\PMYLS ... northwind 00:00:00 3 rows

12.2 Catalog View

Catalog Views are categorized into various groups also. These are used to show the self-describing information of a database. These start with “**sys**”.

This query provides information to all types of views using a database.

