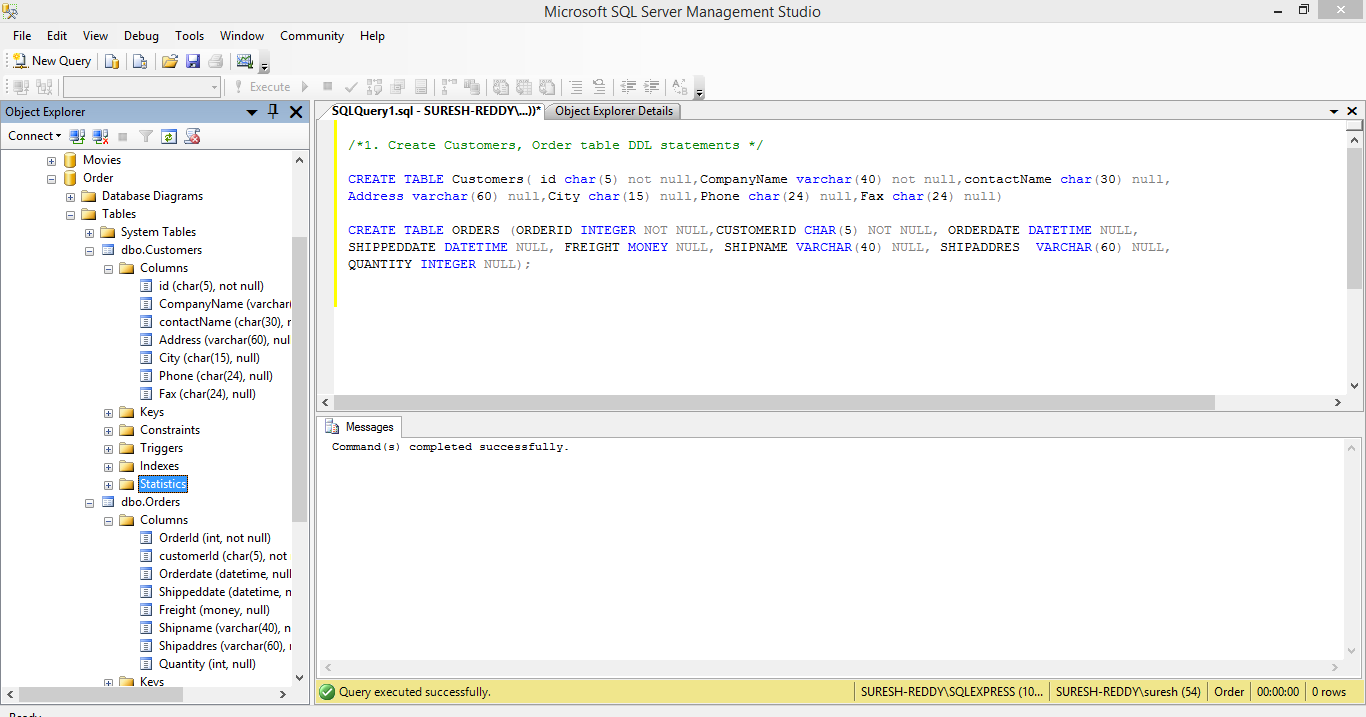
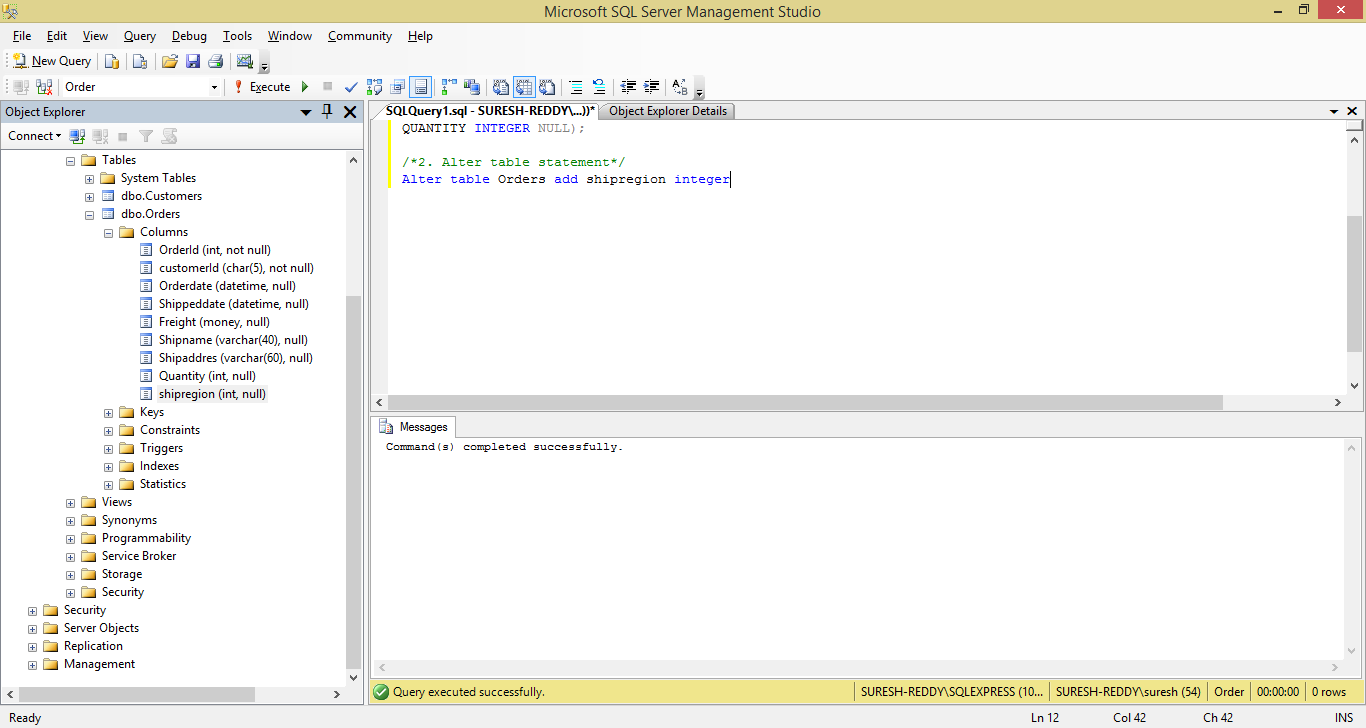
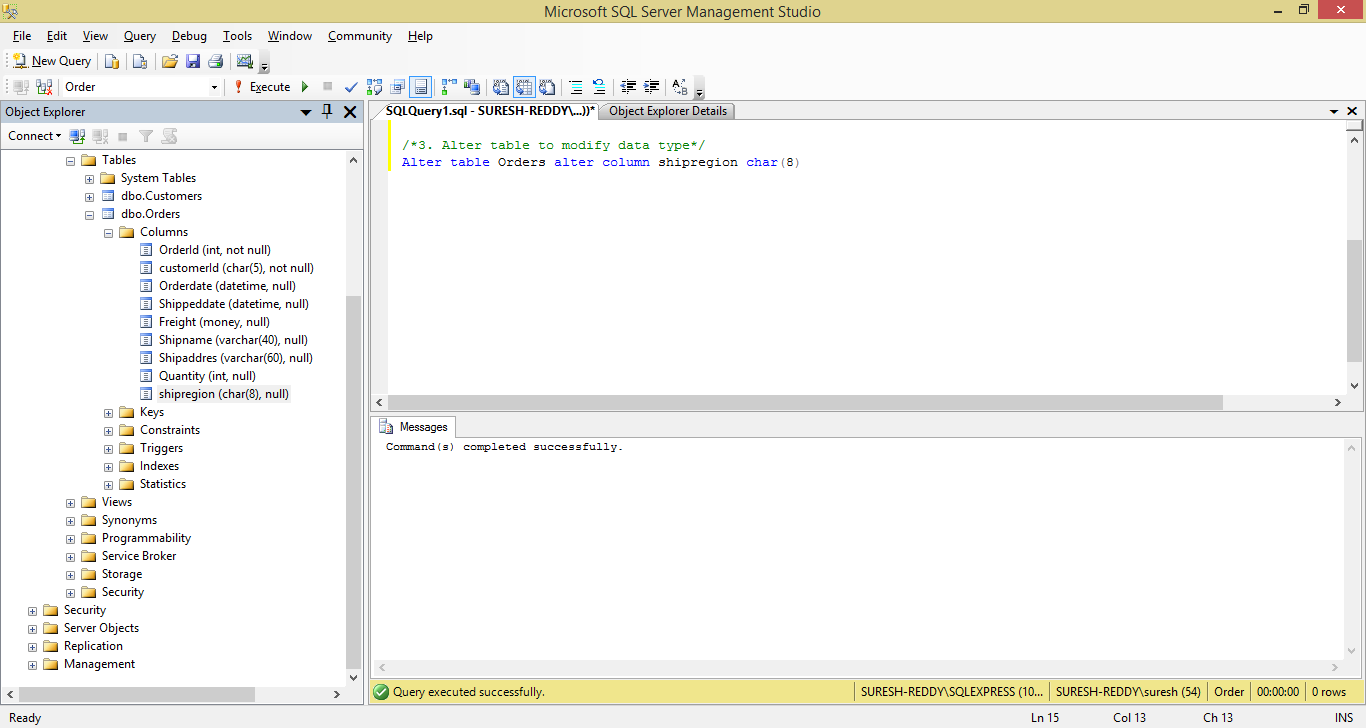
1. Create Customers and Order tables DDL statements



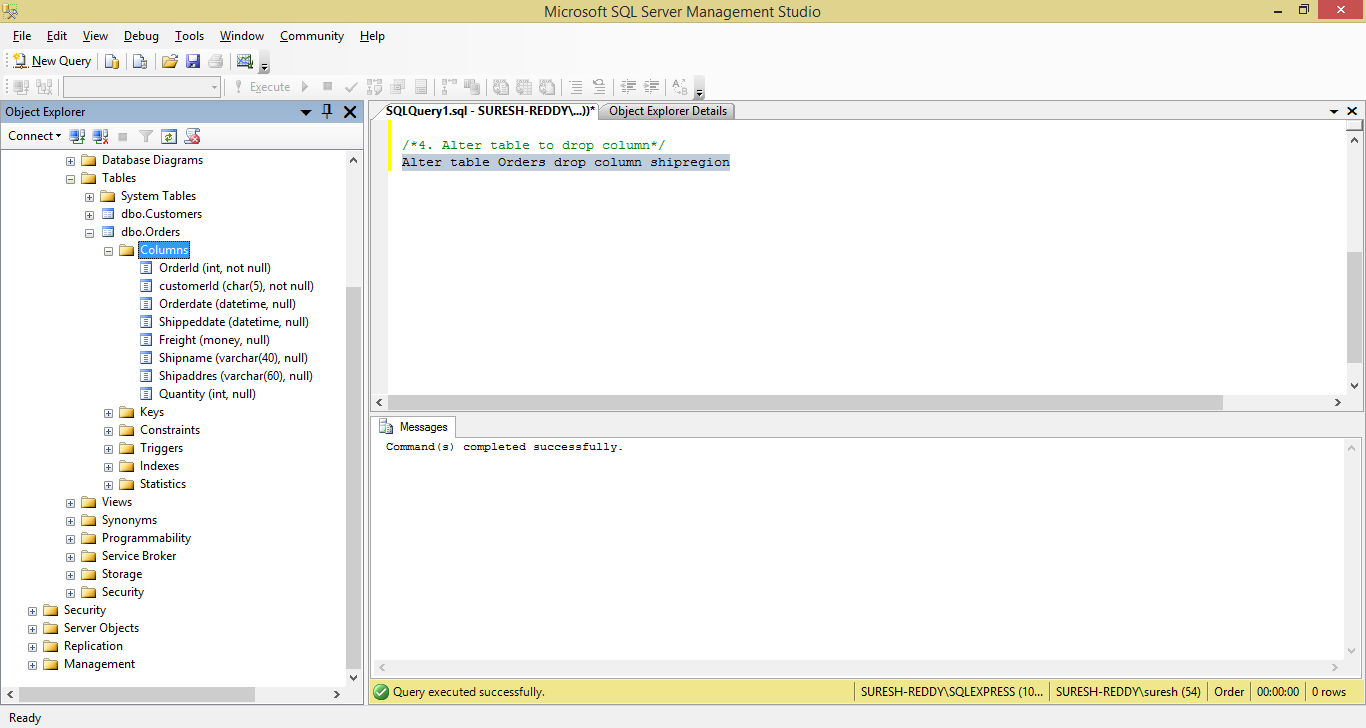
1. Alter Orders table statement



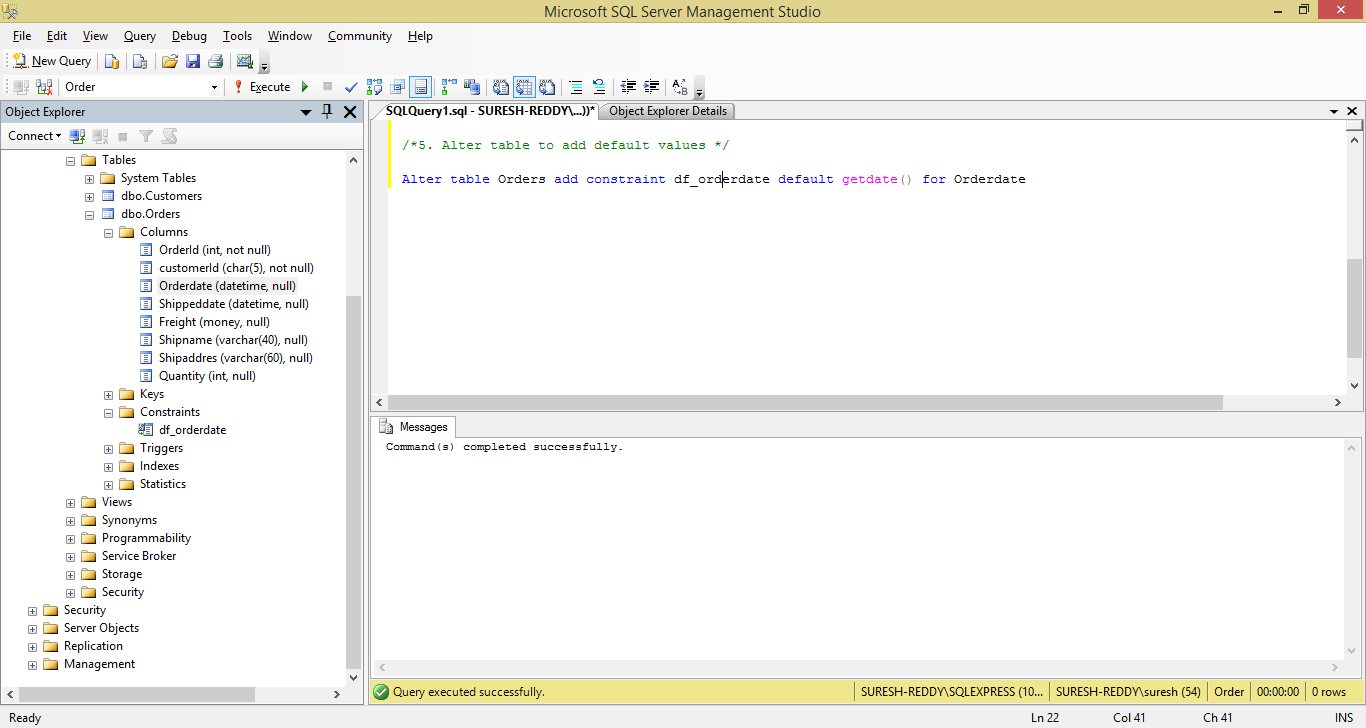
3. Alter table to modify data type



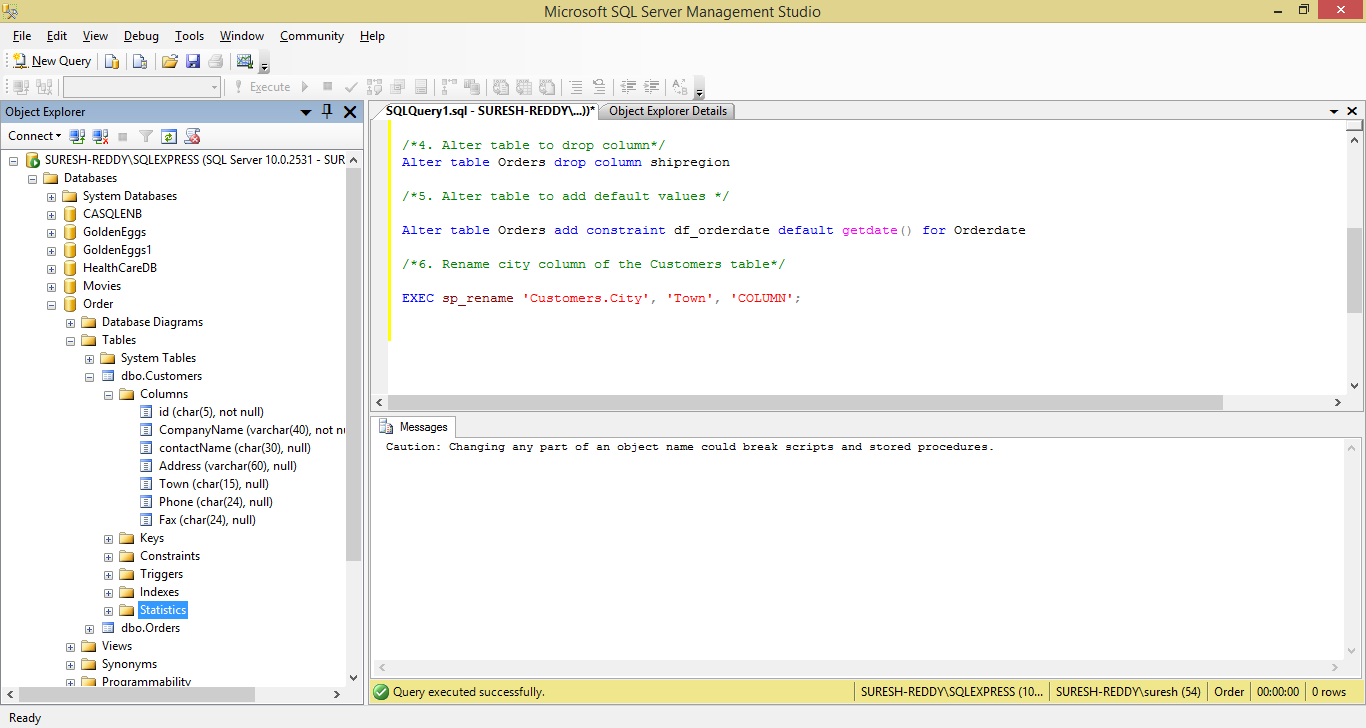
4. Alter table to drop column



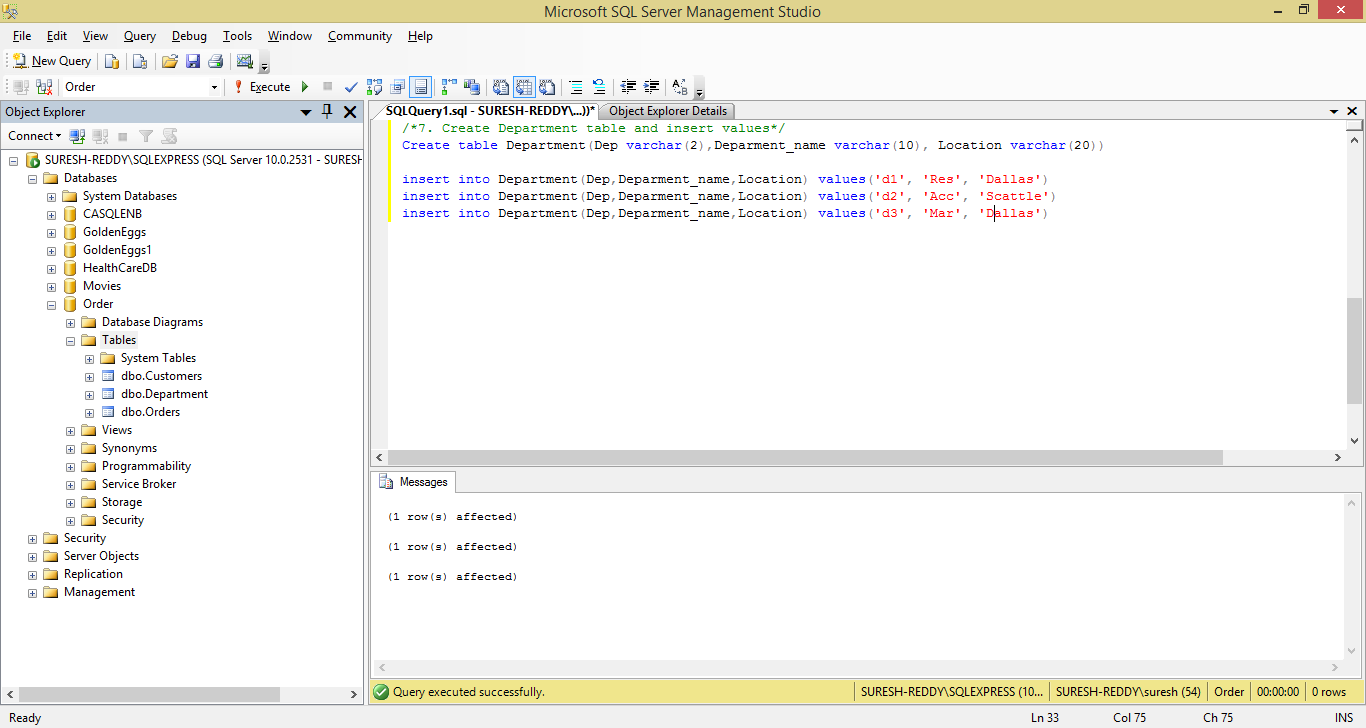
5. Alter statement to add default value



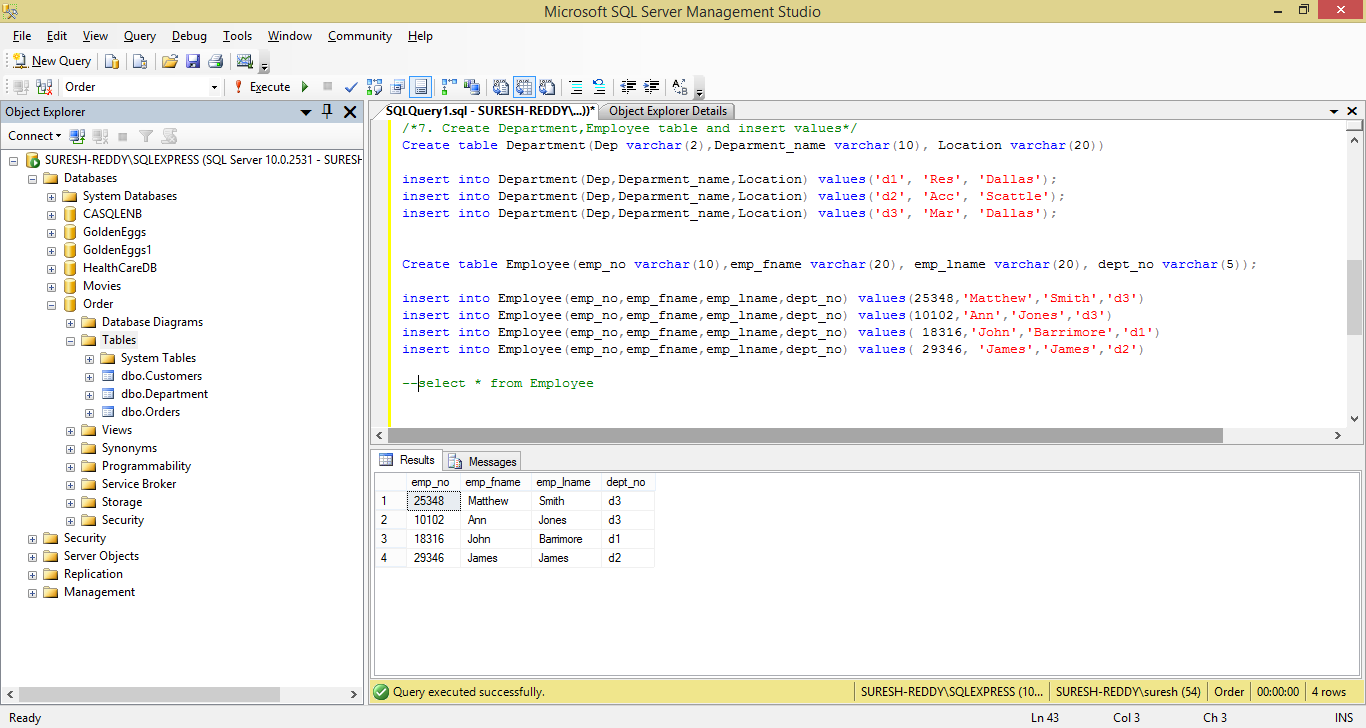
6. Rename the column of Customers table



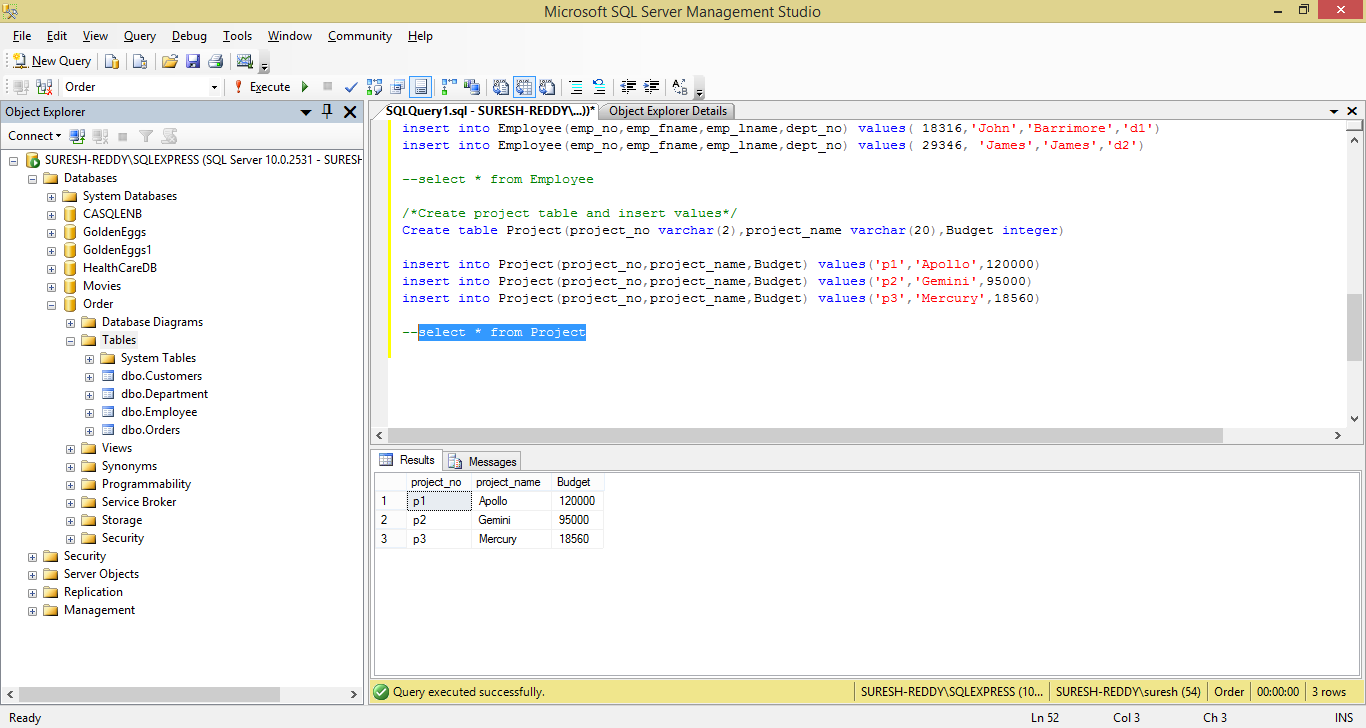
7. Create department and insert values



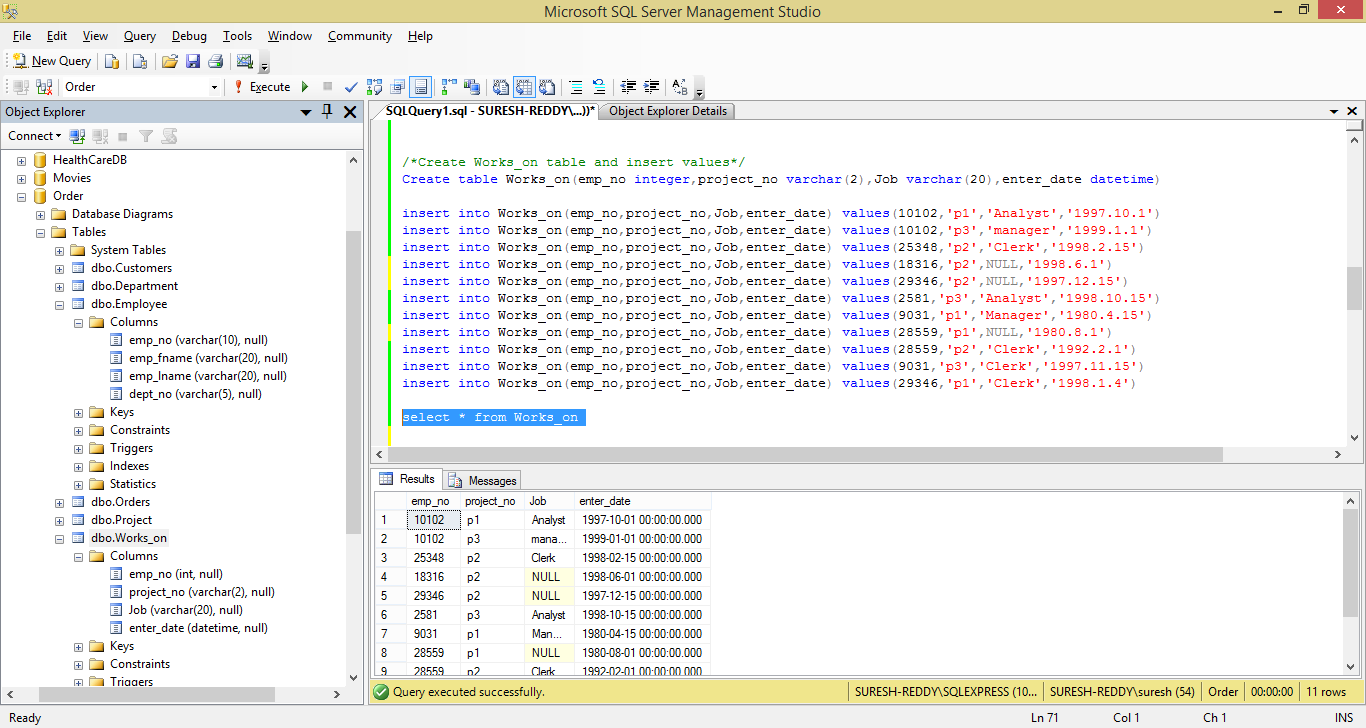
Create Employee table and Insert table



Create Project Table and Insert values



Create Works\_on table and insert table



Simple Queries:-

1. Get the employee numbers for employees working in project p2, and having employee numbers smaller than 10000. Solve this problem with two different but equivalent SELECT statements

SELECT EMP.\* FROM

EMPLOYEE EMP INNER JOIN WORKS\_ON WORKS

ON EMP.EMP\_NO=WORKS.EMP\_NO

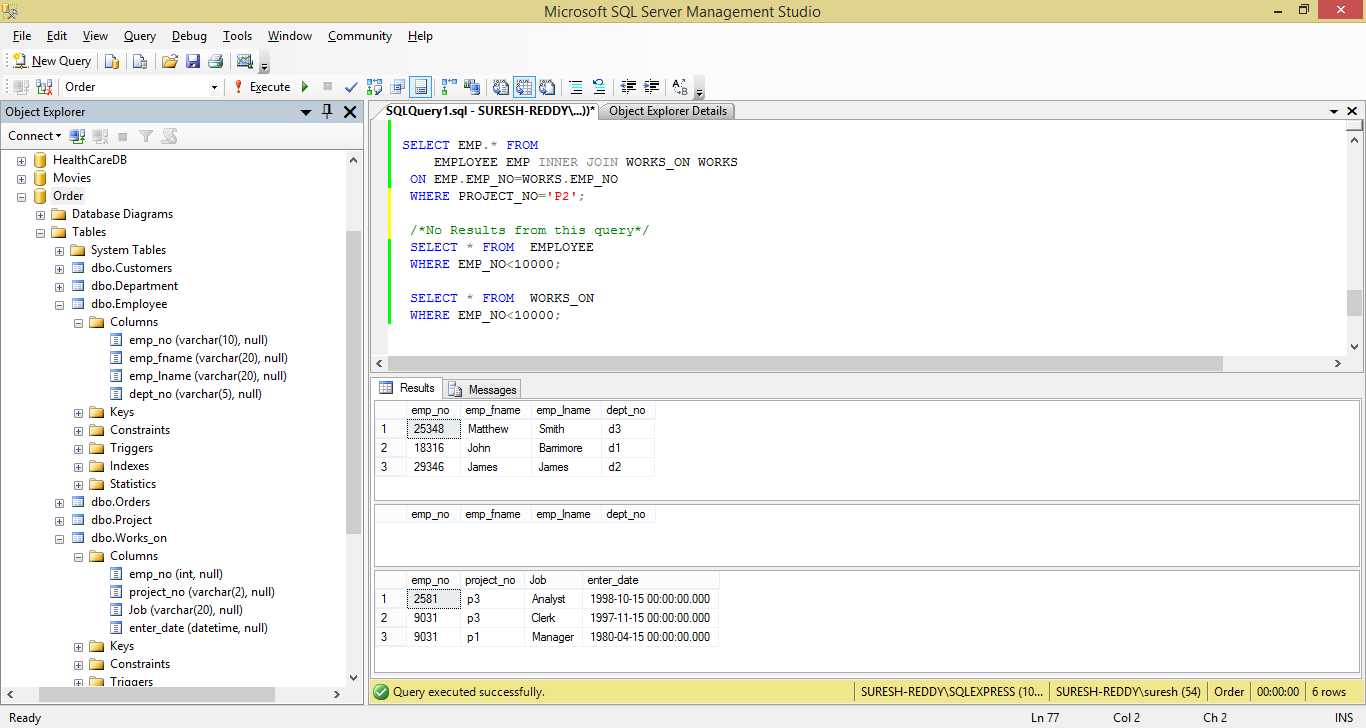
WHERE PROJECT\_NO='P2';

/\*No Results from this query\*/

SELECT \* FROM EMPLOYEE

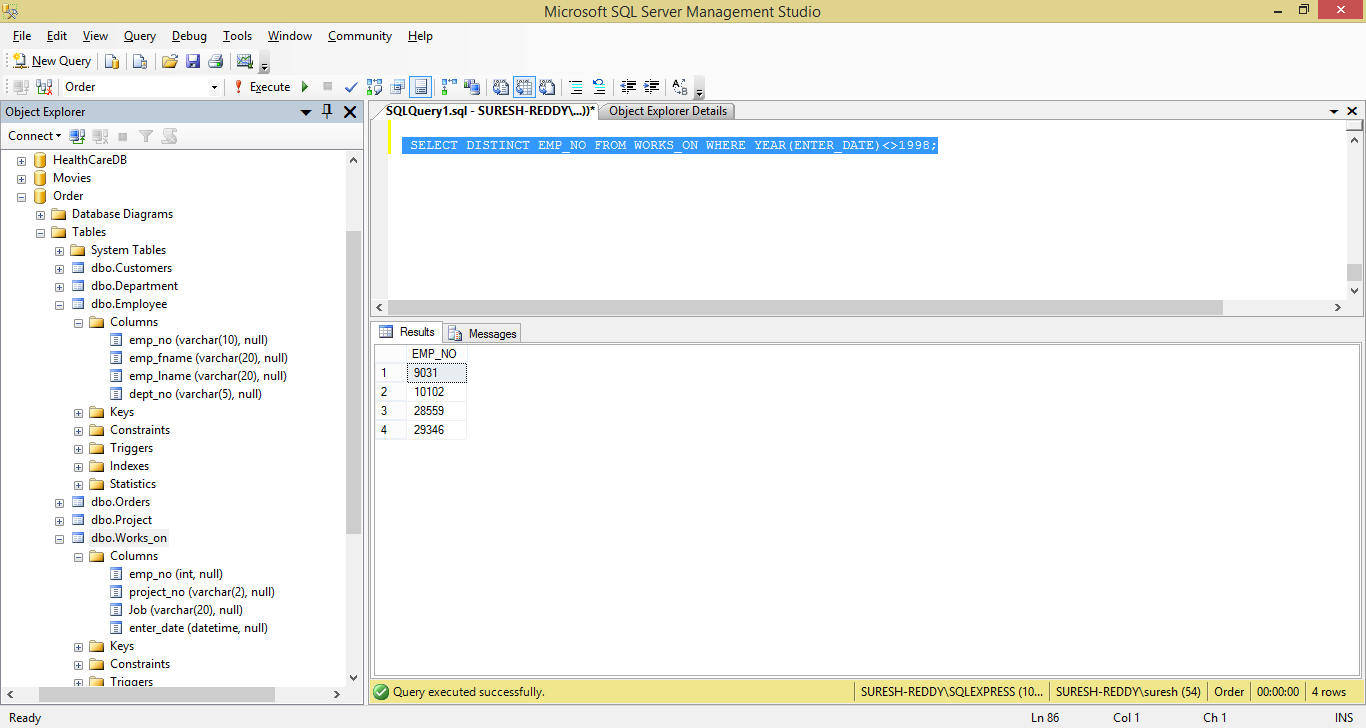
WHERE EMP\_NO<10000;

SELECT \* FROM WORKS\_ON

WHERE EMP\_NO<10000;

1. Get the employee numbers for all employees who didn’t enter their project in 1998

SELECT DISTINCT EMP\_NO FROM WORKS\_ON WHERE YEAR(ENTER\_DATE)<>1998;



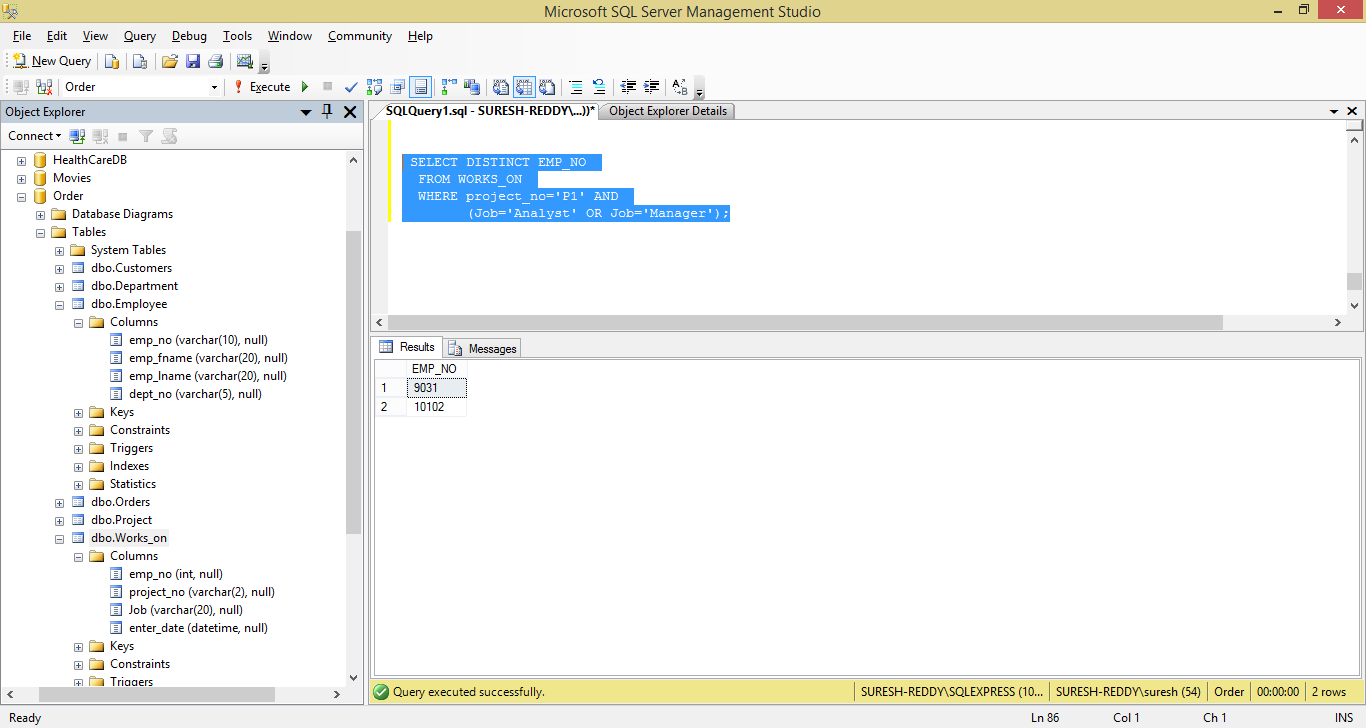
1. Get the employee numbers for all employees who have a leading job( i.e., Analyst or Manager) in project p14

SELECT DISTINCT EMP\_NO

FROM WORKS\_ON

WHERE project\_no='P1' AND

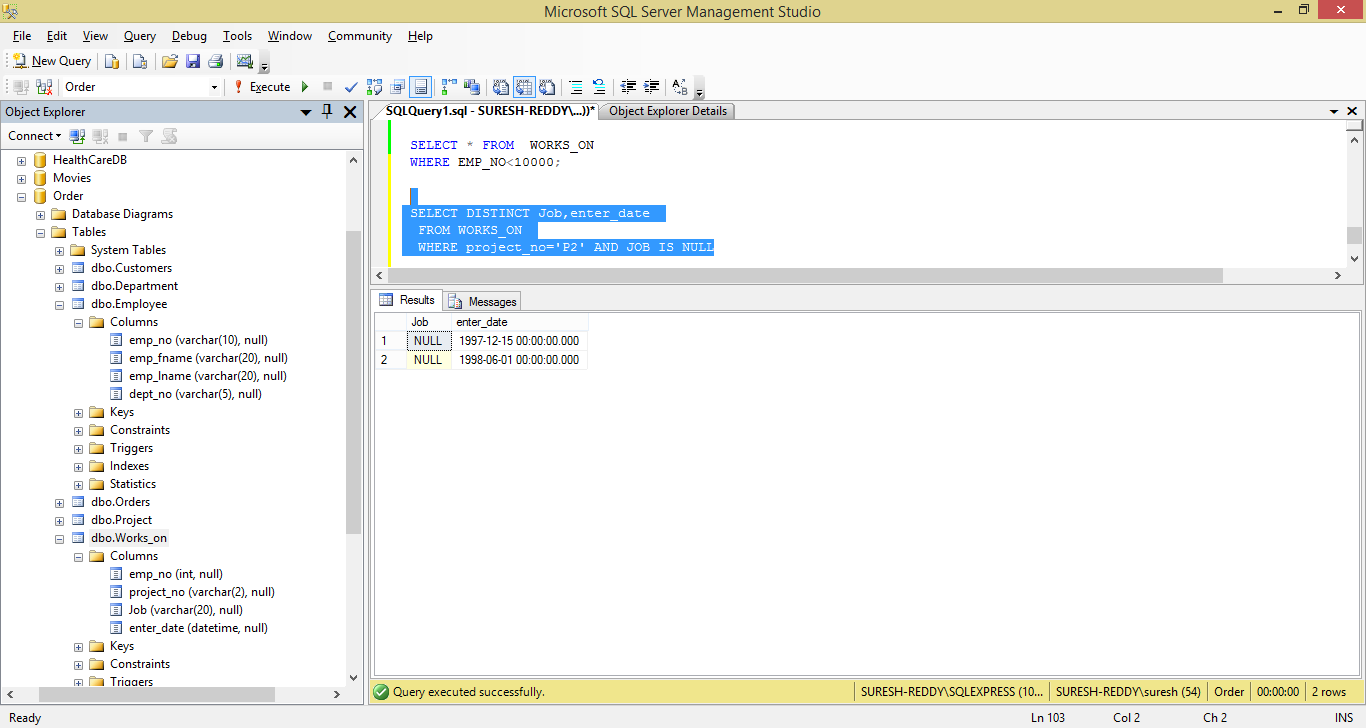
(Job='Analyst' OR Job='Manager');



1. Get the enter dates for all employess in project p2 whose jobs have not been determined yet

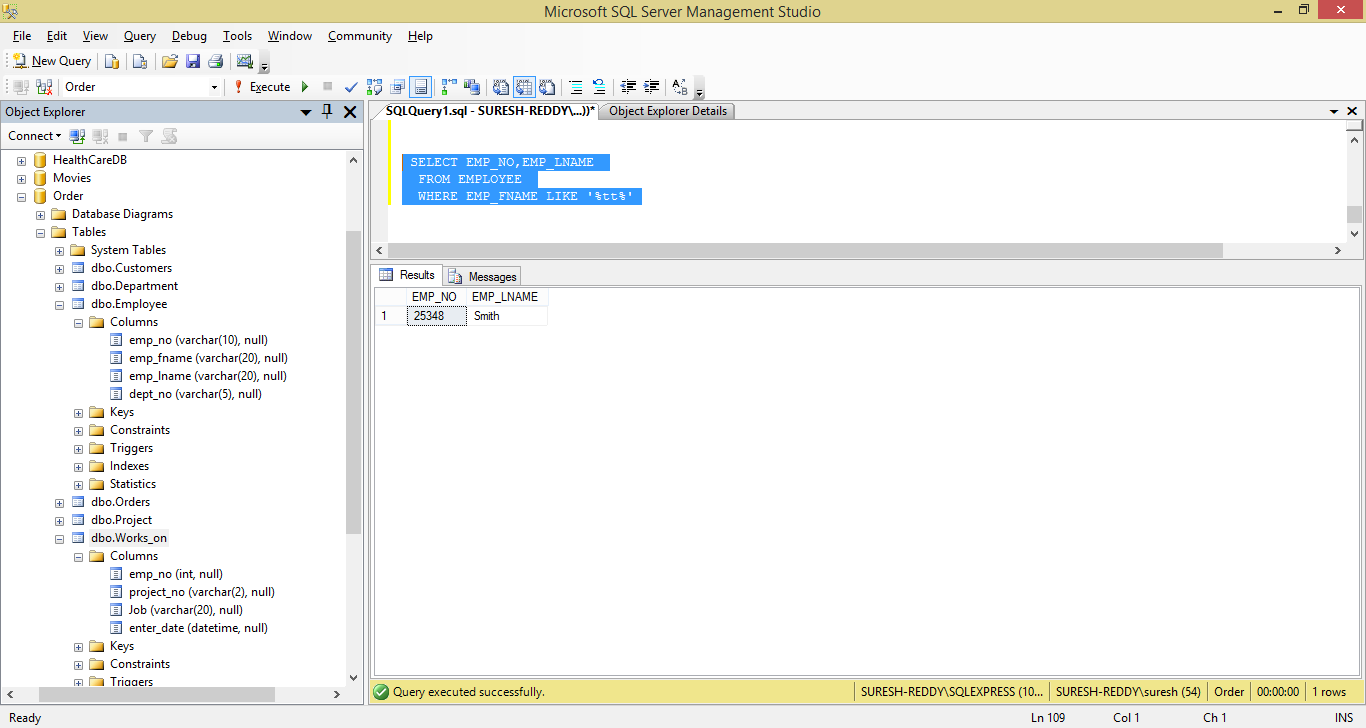
SELECT DISTINCT Job,enter\_date

FROM WORKS\_ON WHERE project\_no='P2' AND JOB IS NULL



1. Get the employee numbers and last names of all employees whose first names contain two letter t’s

SELECT EMP\_NO,EMP\_LNAME FROM EMPLOYEE WHERE EMP\_FNAME LIKE '%tt%'

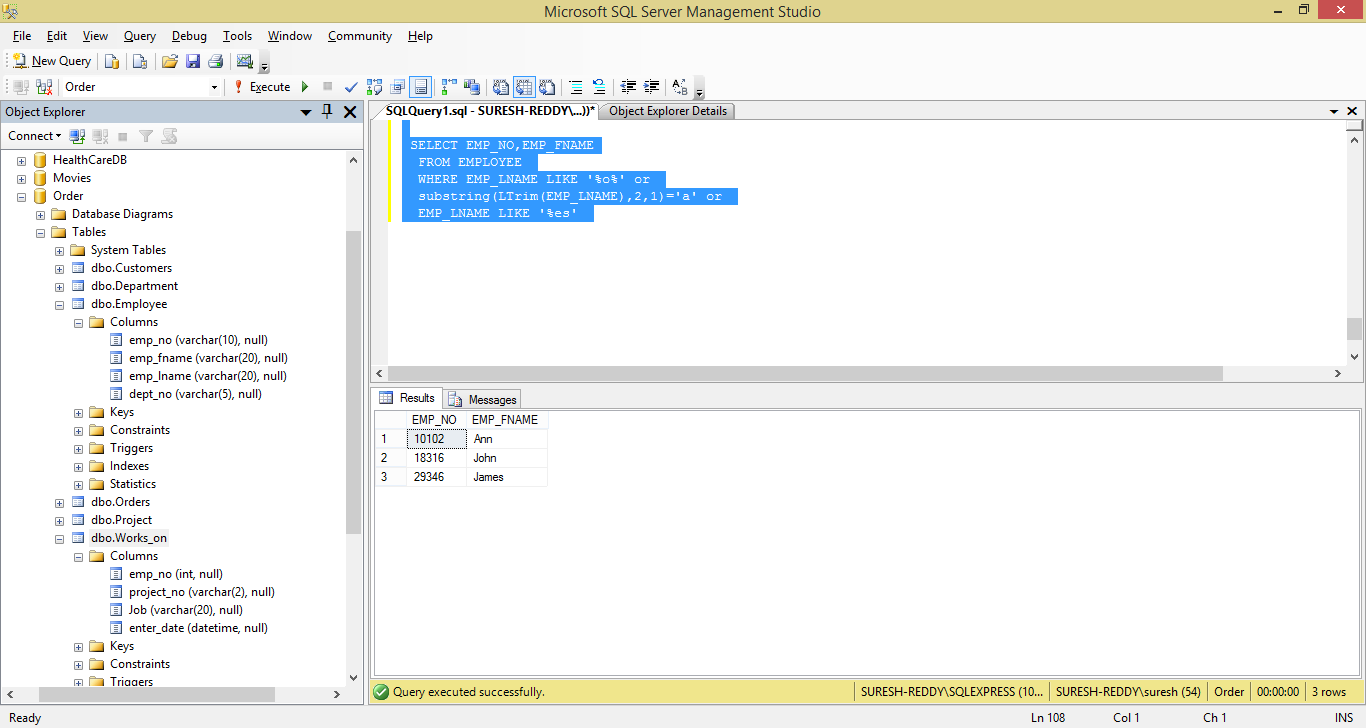


1. Get the employee numbers and first names of all employees whose last names have a letter o or a as the second character and end with the letters es

SELECT EMP\_NO,EMP\_FNAME FROM EMPLOYEE

WHERE EMP\_LNAME LIKE '%o%' or

substring(LTrim(EMP\_LNAME),2,1)='a' or EMP\_LNAME LIKE '%es'



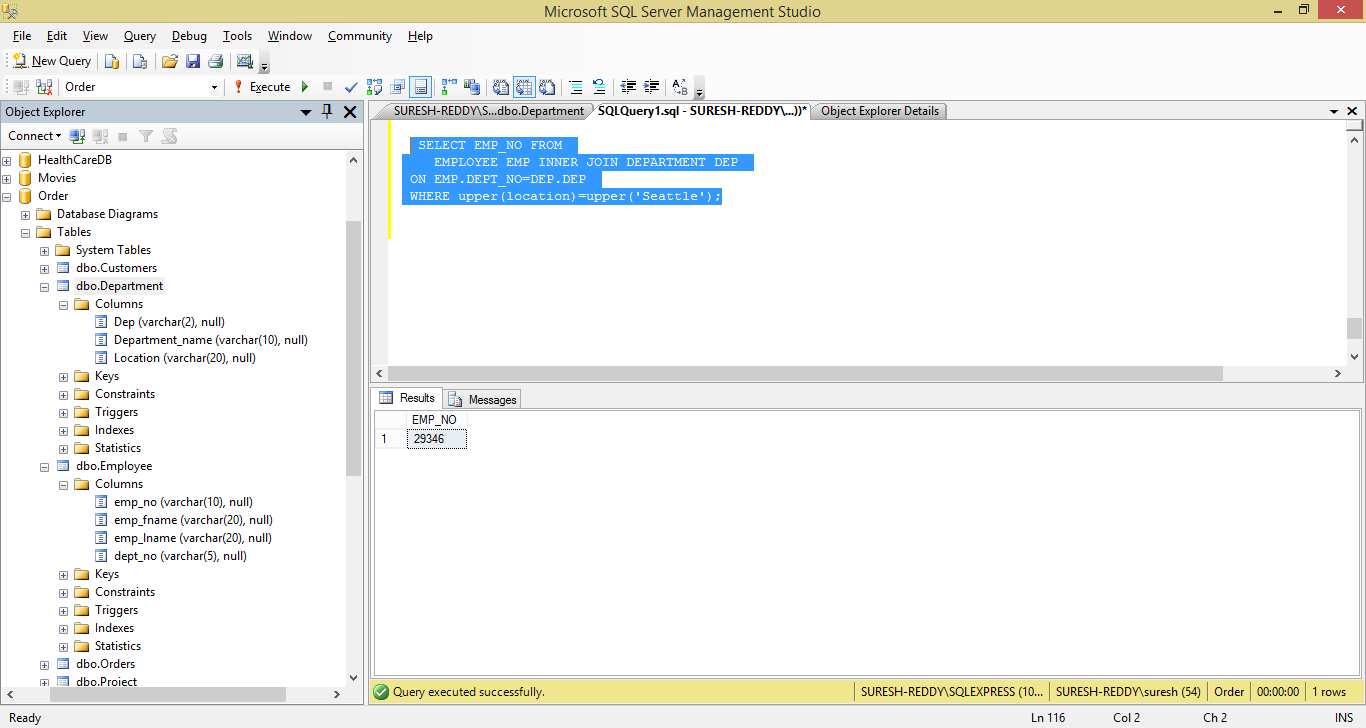
1. Get the employee numbers of all employees whose departments are located in Seattle

SELECT EMP\_NO FROM

EMPLOYEE EMP INNER JOIN DEPARTMENT DEP

ON EMP.DEPT\_NO=DEP.DEP

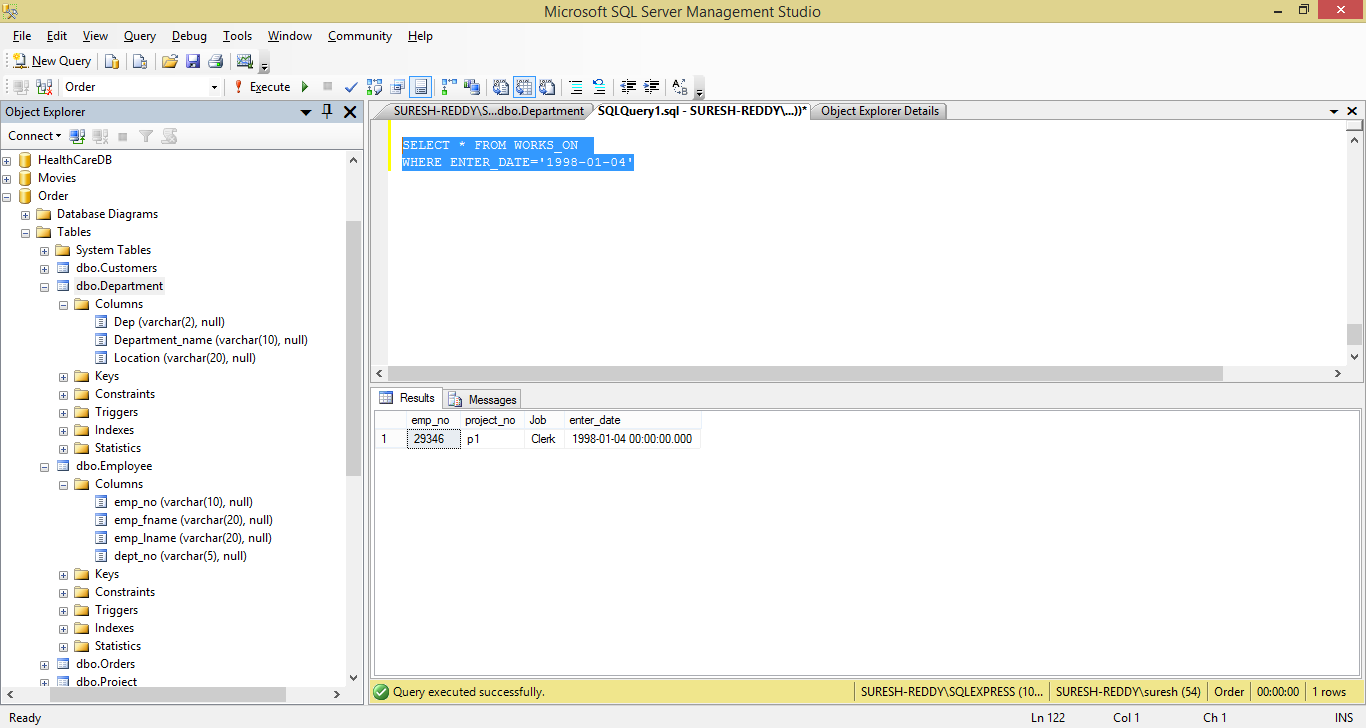
WHERE upper(location)=upper('Seattle');



1. Find the last and first names of all employess who entered their projectson 04.01.1998

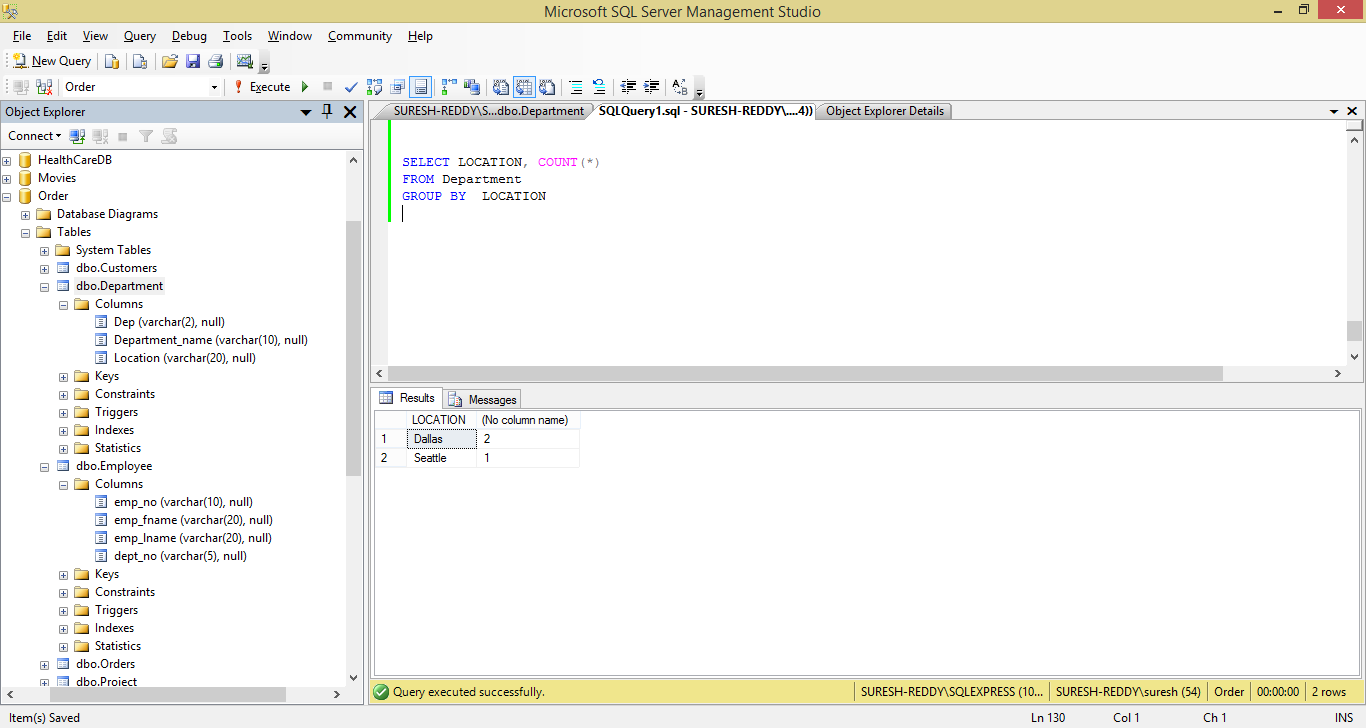
SELECT \* FROM WORKS\_ON

WHERE ENTER\_DATE='1998-01-04'



1. Group all departments using their locations

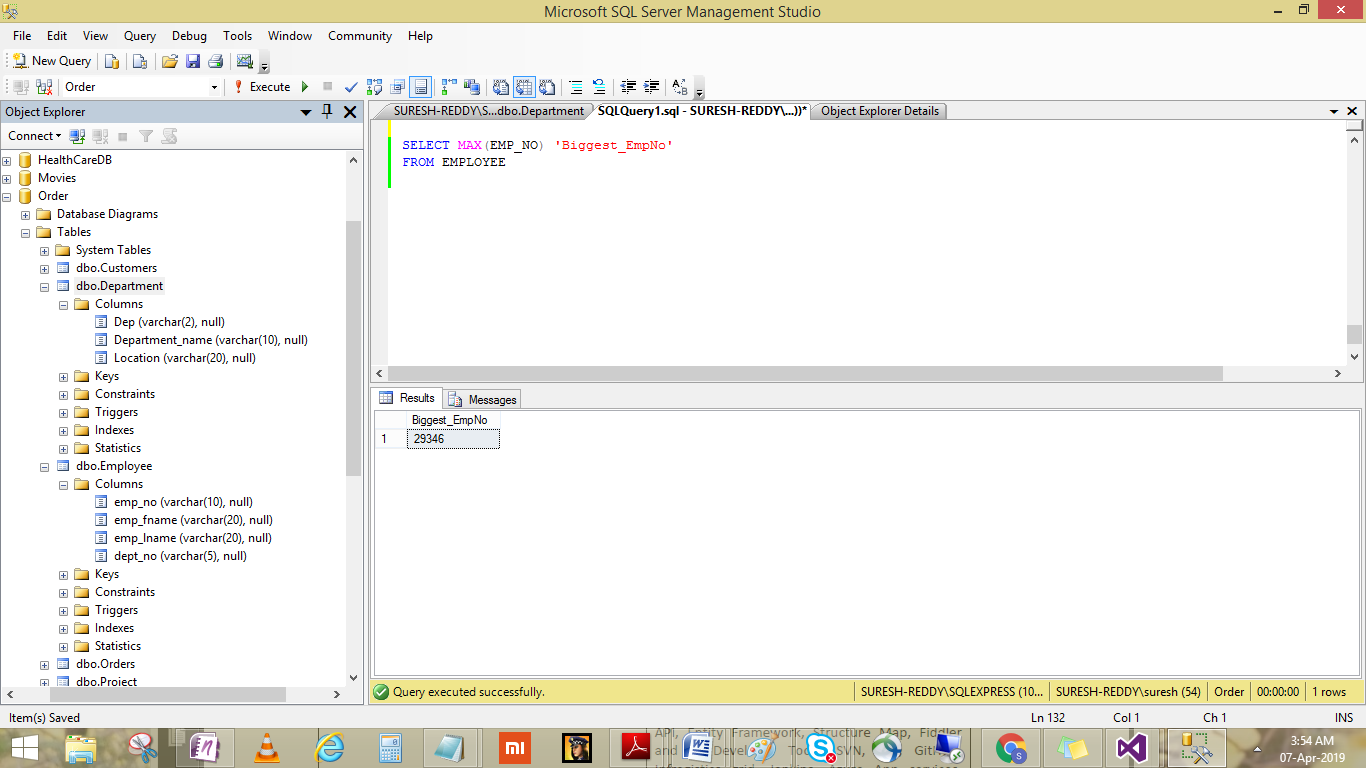
SELECT LOCATION, COUNT(\*) FROM Department GROUP BY LOCATION



1. Find the biggest employee number

SELECT MAX(EMP\_NO) 'Biggest\_EmpNo'

FROM EMPLOYEE



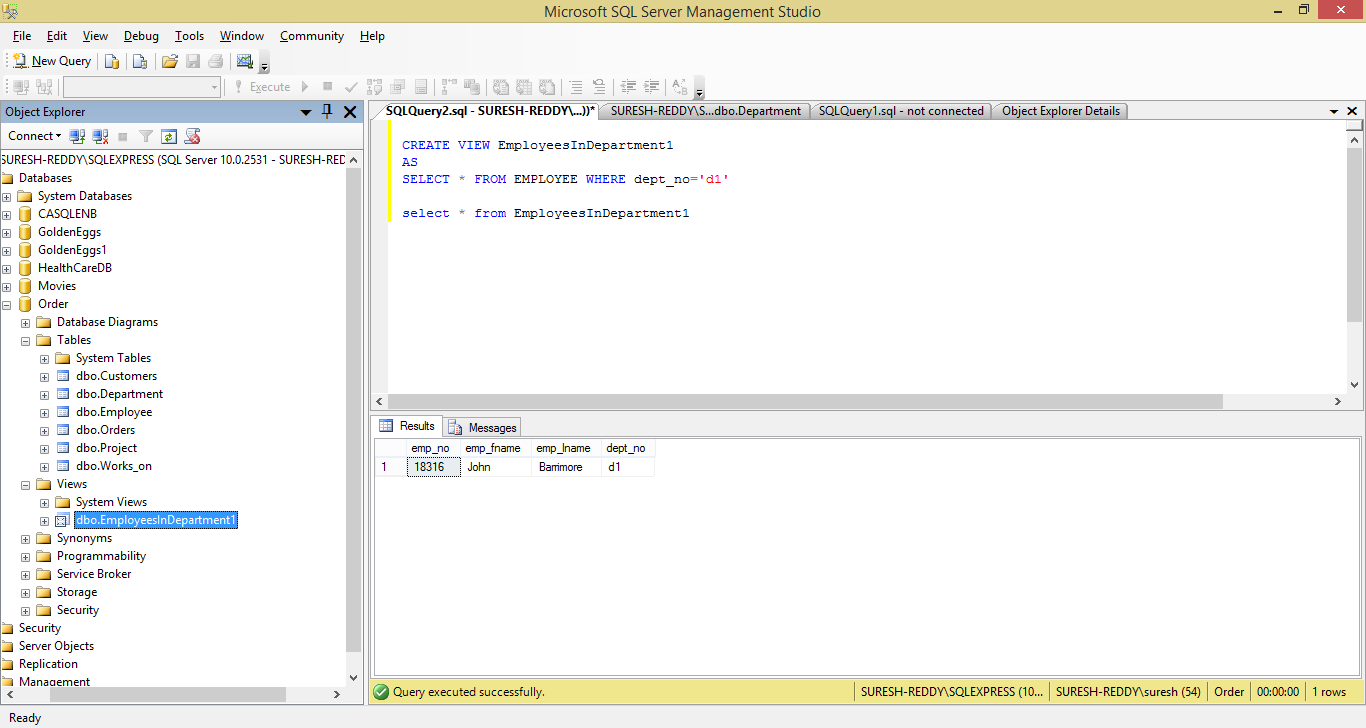
Views:

1. Create a view that comprises the data of all employees that work for the department d1

CREATE VIEW EmployeesInDepartment1

AS

SELECT \* FROM EMPLOYEE WHERE dept\_no='d1'



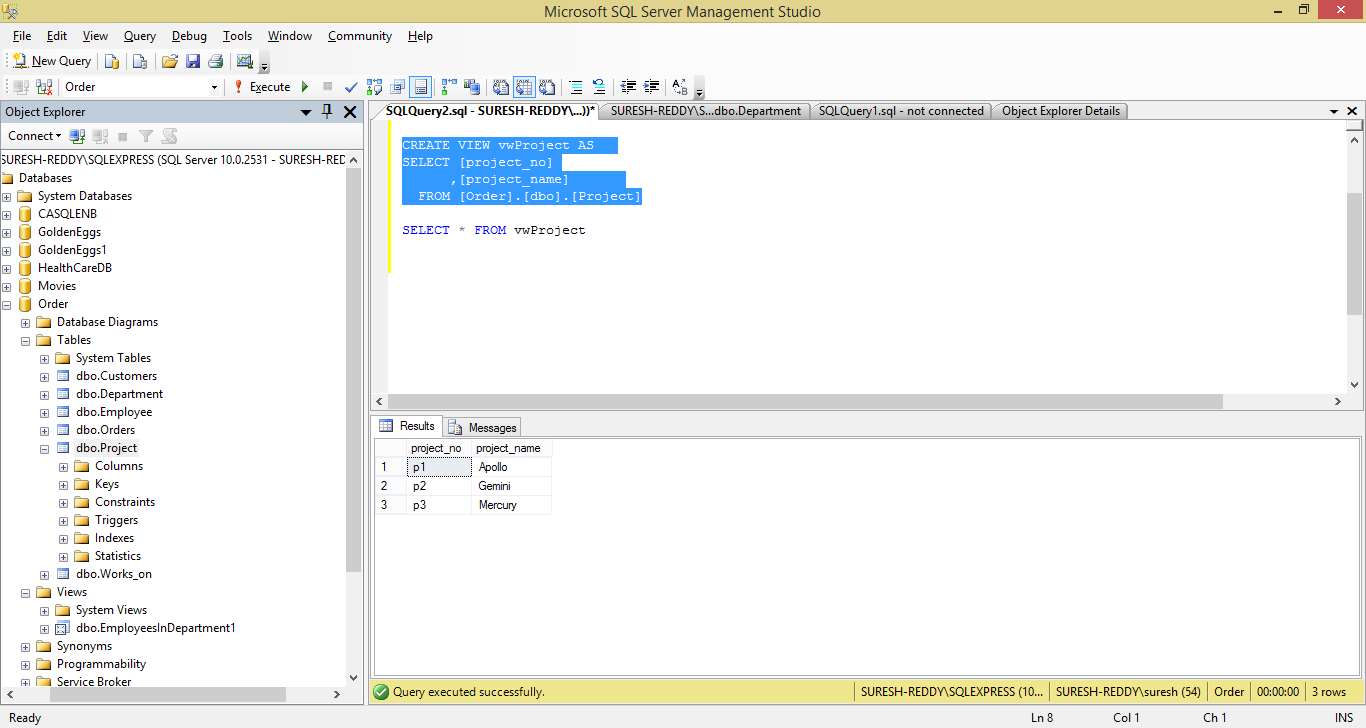
1. For the project table, create a view that can be used by employees who are allowed to view all data of this table except the budget column

CREATE VIEW vwProject AS

SELECT [project\_no]

,[project\_name]

FROM [Order].[dbo].[Project]



1. Create a view that comprises the first and last names of all employees who entered their projects in the second half of the year 1988

CREATE VIEW vmEmployeesEnterdInSecondHalf

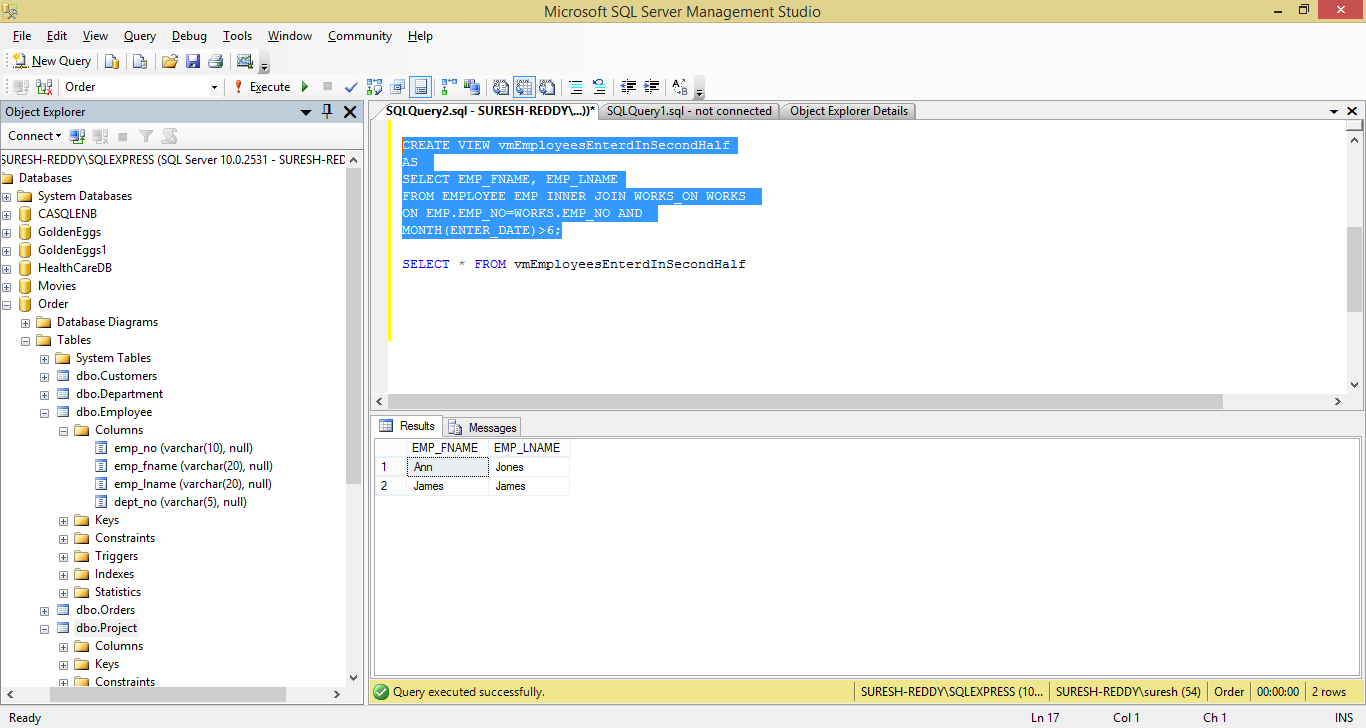
AS

SELECT EMP\_FNAME, EMP\_LNAME

FROM EMPLOYEE EMP INNER JOIN WORKS\_ON WORKS

ON EMP.EMP\_NO=WORKS.EMP\_NO AND

MONTH(ENTER\_DATE)>6;



**Constraints**:-

1. Re-create the Customers and Orders tables, enhancing their definition with all primary and foreign keys constraints

--drop table Customers

CREATE TABLE Customers( id char(5) not null primary key,CompanyName varchar(40) not null,contactName char(30) null,

Address varchar(60) null,City char(15) null,Phone char(24) null,Fax char(24) null)

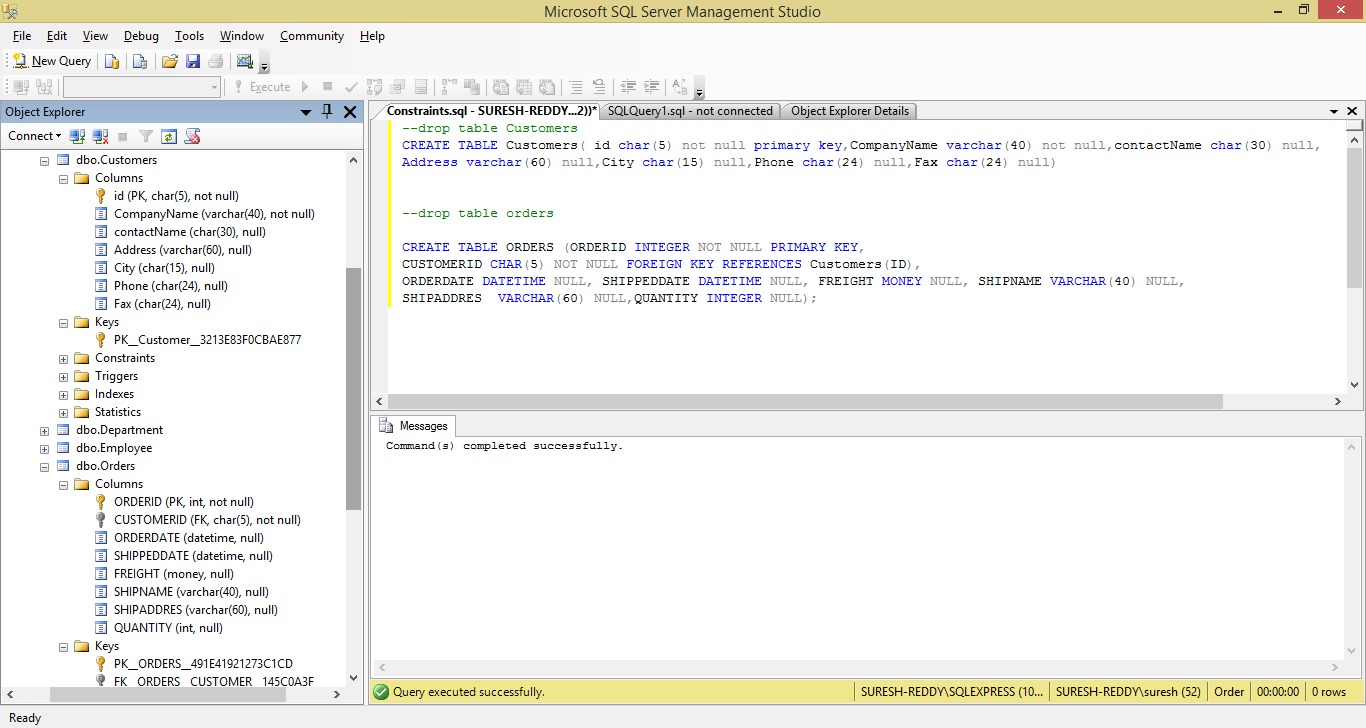
--drop table orders

CREATE TABLE ORDERS (ORDERID INTEGER NOT NULL PRIMARY KEY,

CUSTOMERID CHAR(5) NOT NULL FOREIGN KEY REFERENCES Customers(ID),

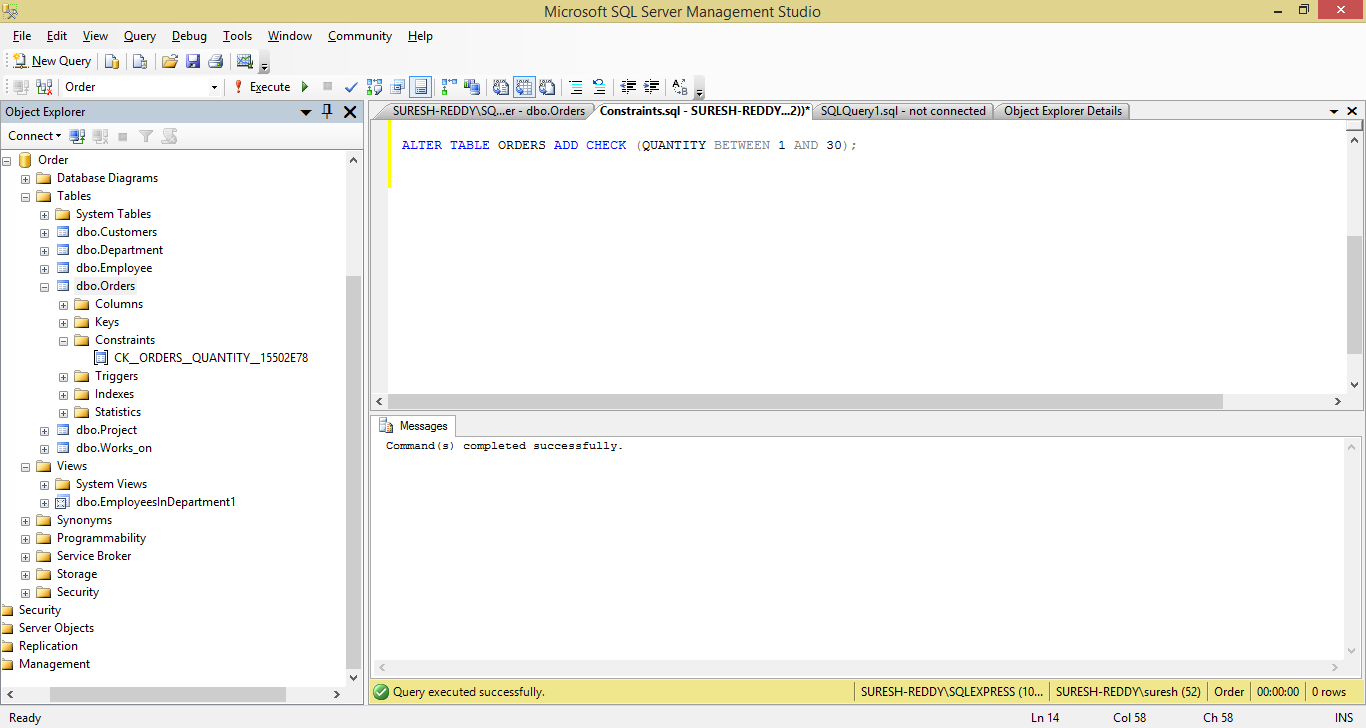
ORDERDATE DATETIME NULL, SHIPPEDDATE DATETIME NULL, FREIGHT MONEY NULL, SHIPNAME VARCHAR(40) NULL,

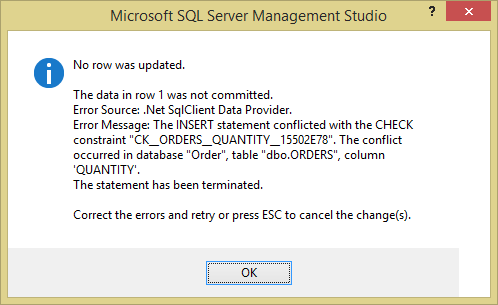
SHIPADDRES VARCHAR(60) NULL,QUANTITY INTEGER NULL);



2. Using the ALTER TABLE statement, create an integrity constraint that limits the possible values of the quantity column in the Orders table to values between 1 and 30

ALTER TABLE ORDERS ADD CHECK (QUANTITY BETWEEN 1 AND 30);





1. .With the help of the corresponding system procedures and the Transact-SQL statements CREATE DEFAULT and CREATE RULE, create the alias data type “Western Countries”. The possible values for the new data type are CA(for California), WA( for Washington), OR( for Oregon), and NM( for New Mexico). The default value is CA. Finally, create a table called Regions with the columns City and Country using the new data type for the later

CREATE DEFAULT Default\_Country

AS 'CA'

GO;

CREATE RULE rule\_Country

AS

(@country='Default\_Country')

OR @country in ('CA','WA','OR','NM')

GO;

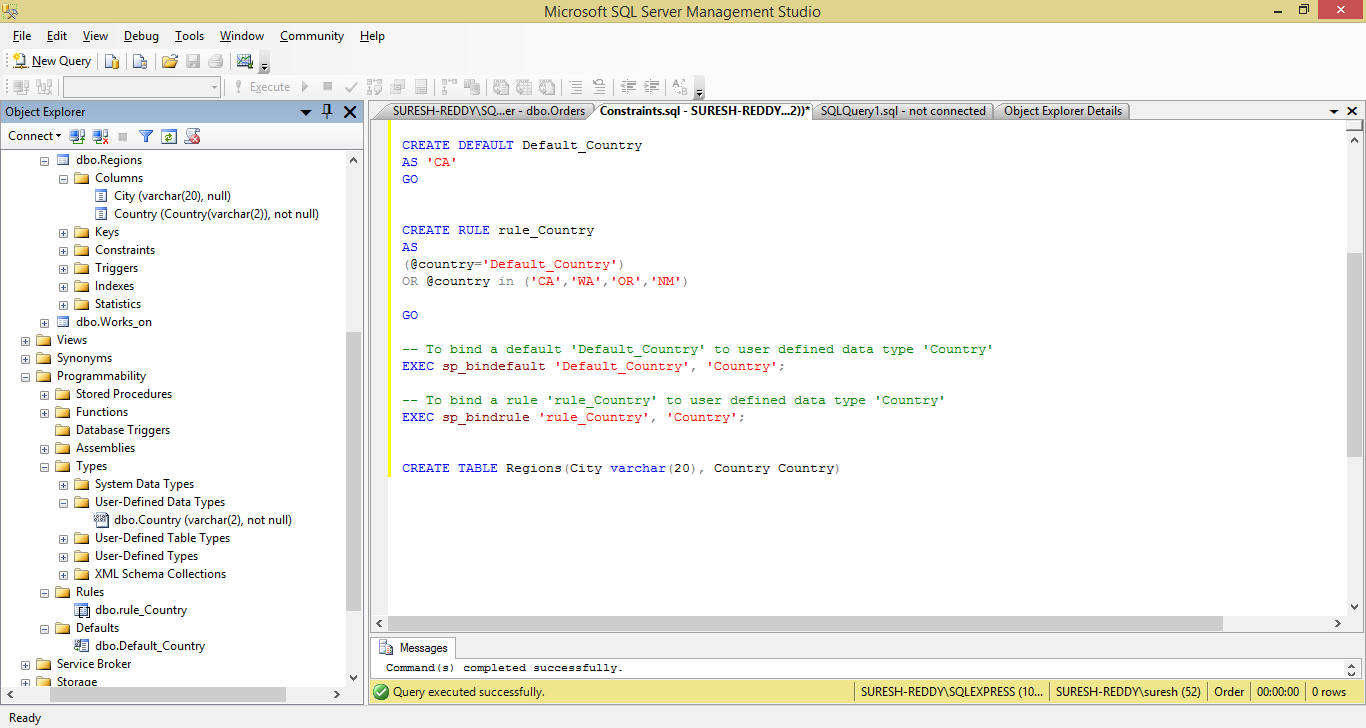
-- To bind a default 'Default\_Country' to user defined data type 'Country'

EXEC sp\_bindefault 'Default\_Country', 'Country';

-- To bind a rule 'rule\_Country' to user defined data type 'Country'

EXEC sp\_bindrule 'rule\_Country', 'Country';

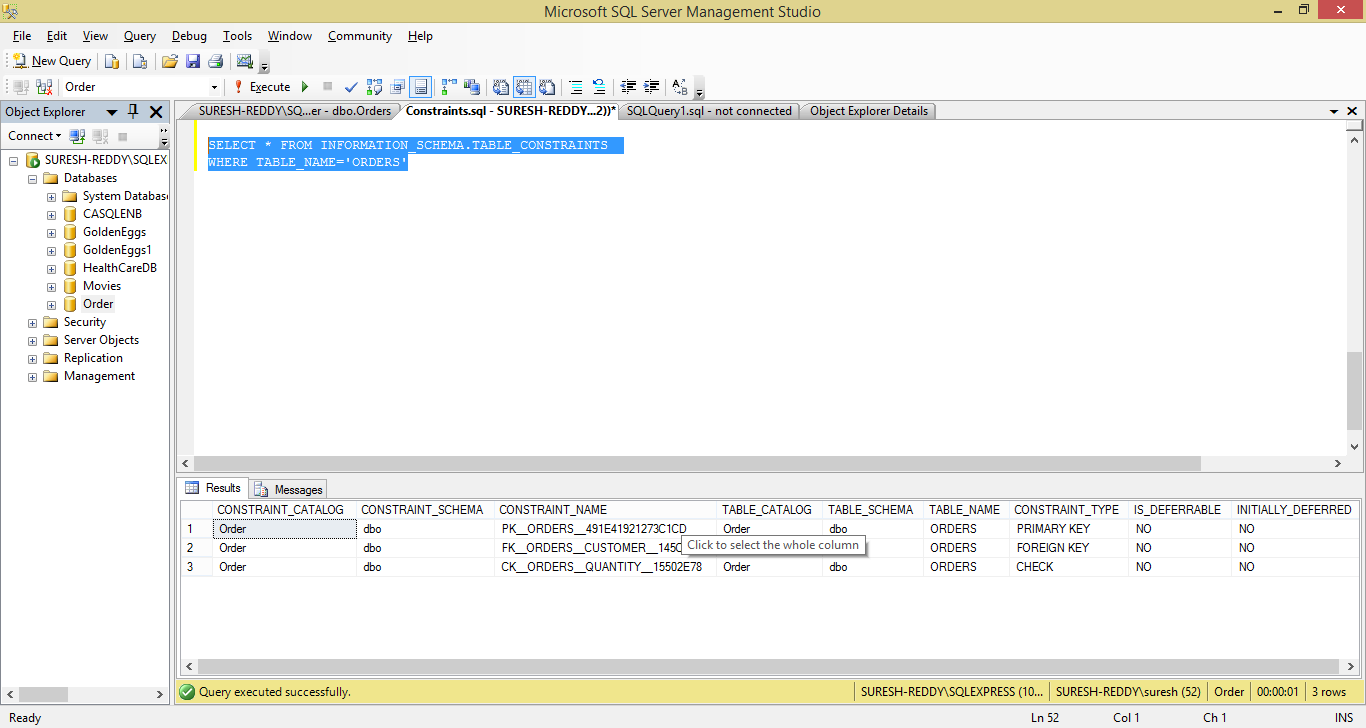
CREATE TABLE Regions(City varchar(20), Country Country);



1. Display all integrity constraints for the Orders table

SELECT \* FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_NAME='ORDERS'



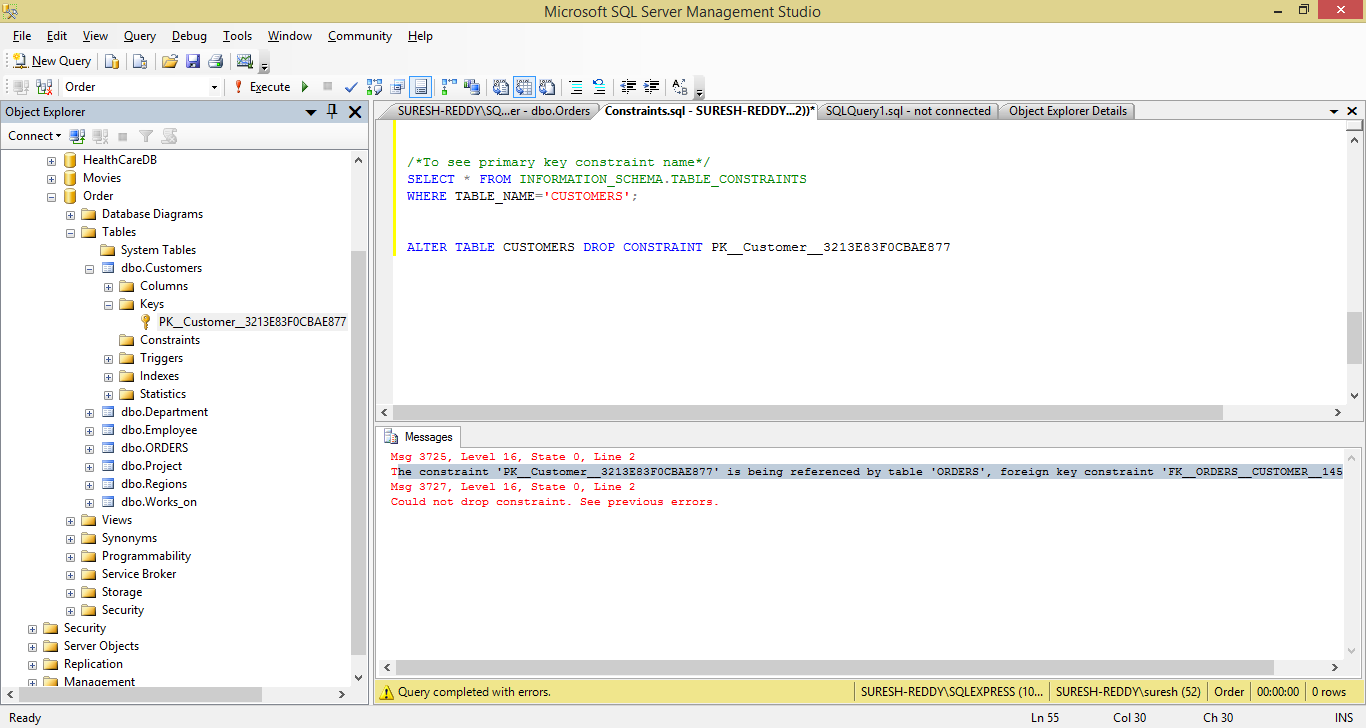
1. Delete the primary key of the Customers table

/\*To see primary key constraint name\*/

SELECT \* FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

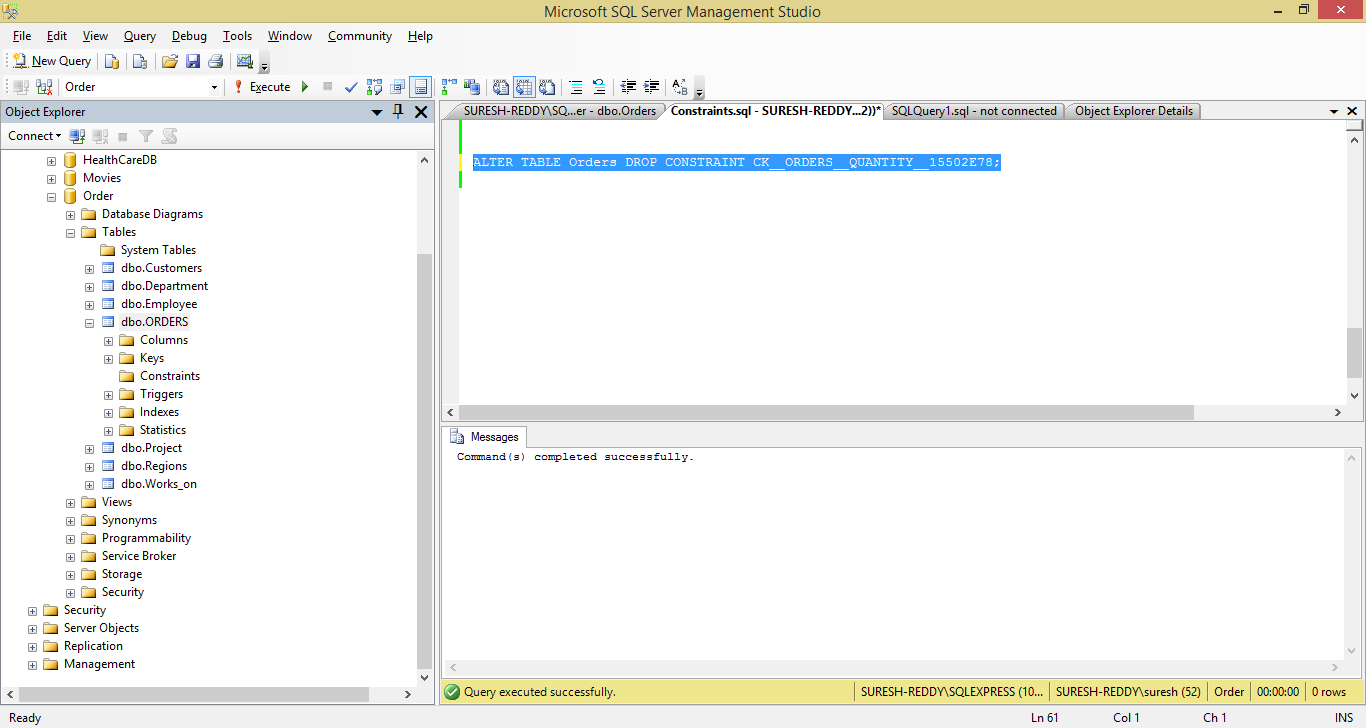
WHERE TABLE\_NAME='CUSTOMERS';

ALTER TABLE CUSTOMERS DROP CONSTRAINT PK\_\_Customer\_\_3213E83F0CBAE877



1. Delete the integrity constraint defined in Step-2

ALTER TABLE Orders DROP CONSTRAINT CK\_\_ORDERS\_\_QUANTITY\_\_15502E78;



**Functions and Procedures:**

1. Write a function that will return the age of the person given his or her date of birth

CREATE FUNCTION FNC\_CALCULATEAGE(@DATE\_OF\_BIRTH DATETIME)

RETURNS INT

AS

BEGIN

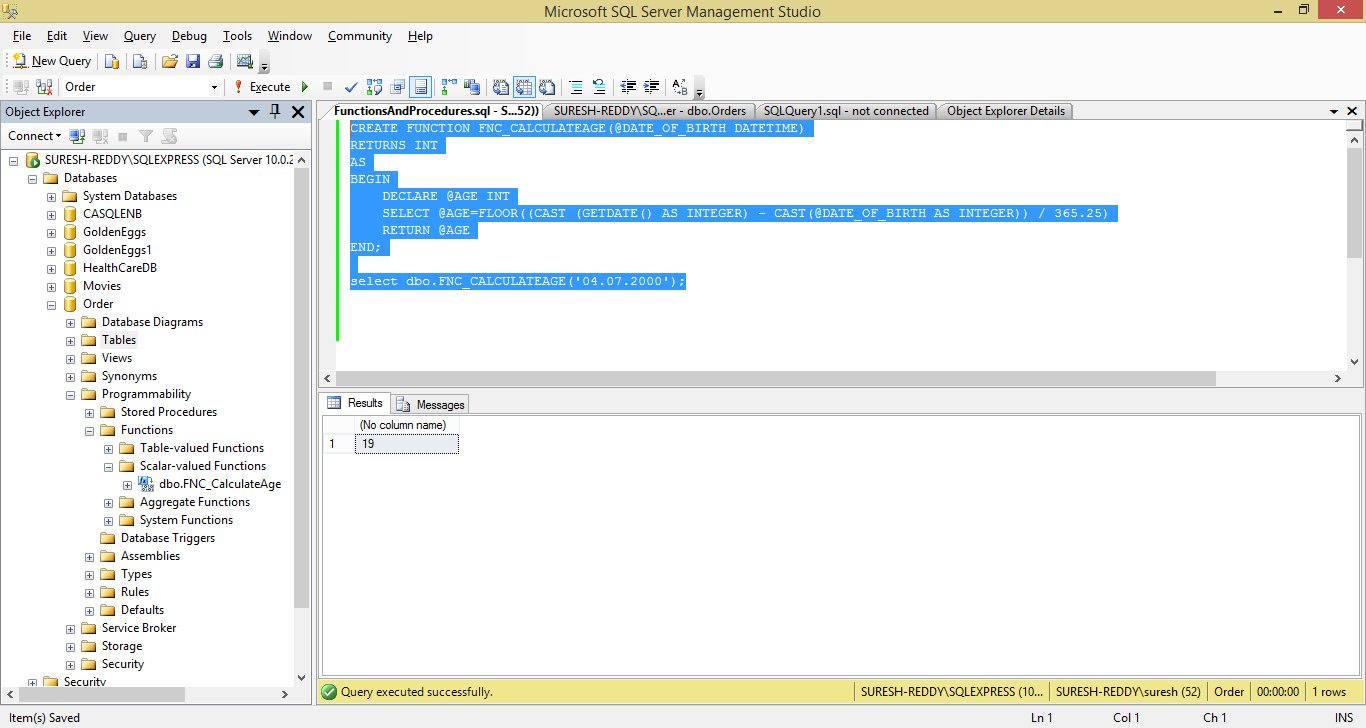
DECLARE @AGE INT

SELECT @AGE=FLOOR((CAST (GETDATE() AS INTEGER) - CAST(@DATE\_OF\_BIRTH AS INTEGER)) / 365.25)

RETURN @AGE

END;

select dbo.FNC\_CALCULATEAGE('04.07.2000');

****

2. Write a procedure that accepts name and data of birth of the student and inserts the data in the student table with the date computed. The SID should be largest sid in the table +1

CREATE TABLE STUDENT(SID INTEGER PRIMARY KEY, NAME VARCHAR(50), DATE\_OF\_BIRTH DATETIME, AGE INT);

/\*Creating store procedure\*/

CREATE PROCEDURE SP\_STUDENT\_INSERT

(@NAME VARCHAR(50),

@DATE\_OF\_BIRTH DATETIME

)

AS

BEGIN

DECLARE @AGE INTEGER

DECLARE @STUDENT\_ID INTEGER=0

SELECT @STUDENT\_ID=isnull(MAX(SID),0)+1 FROM STUDENT

SELECT @AGE=dbo.FNC\_CALCULATEAGE(@DATE\_OF\_BIRTH);

INSERT INTO STUDENT(SID, NAME, DATE\_OF\_BIRTH, AGE)

VALUES(@STUDENT\_ID,@NAME,@DATE\_OF\_BIRTH,@AGE)

END

/\*Insert Rows\*/

EXEC SP\_STUDENT\_INSERT 'Surekha','12.12.2001'

SELECT \* FROM STUDENT;

