**“IN THE NAME OF GOD”**

**Data Encryption in SQL Server**

**Professor :DR Fakhr Ahmad**

PRESENTER: Narges Dehghan

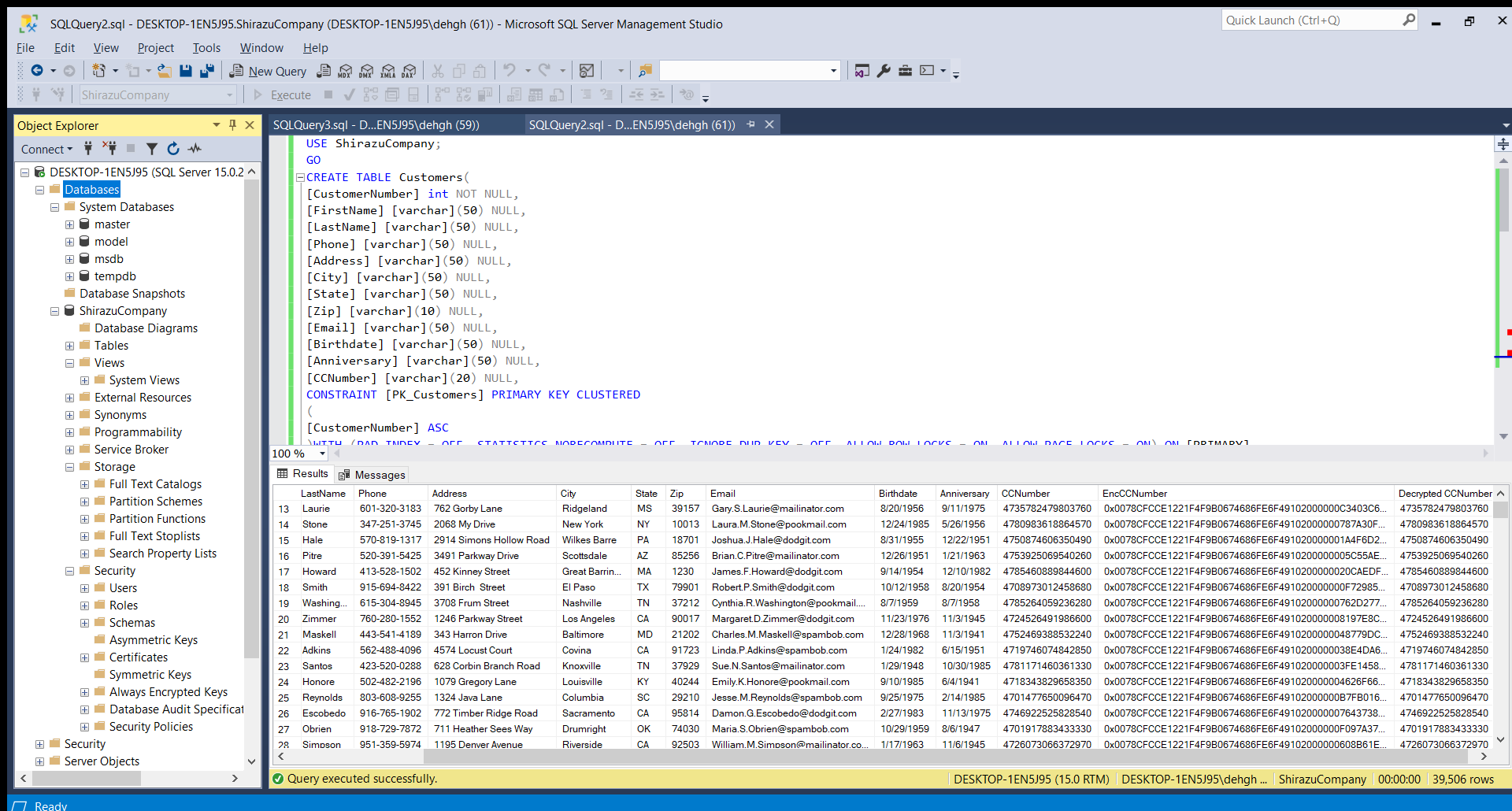
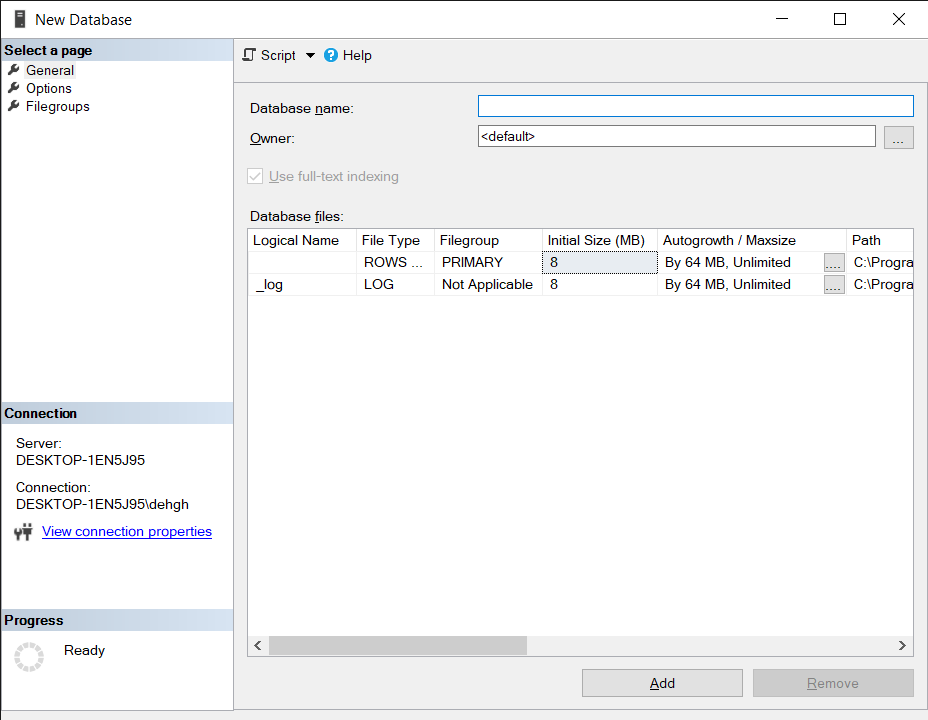
Std.id : 40033756

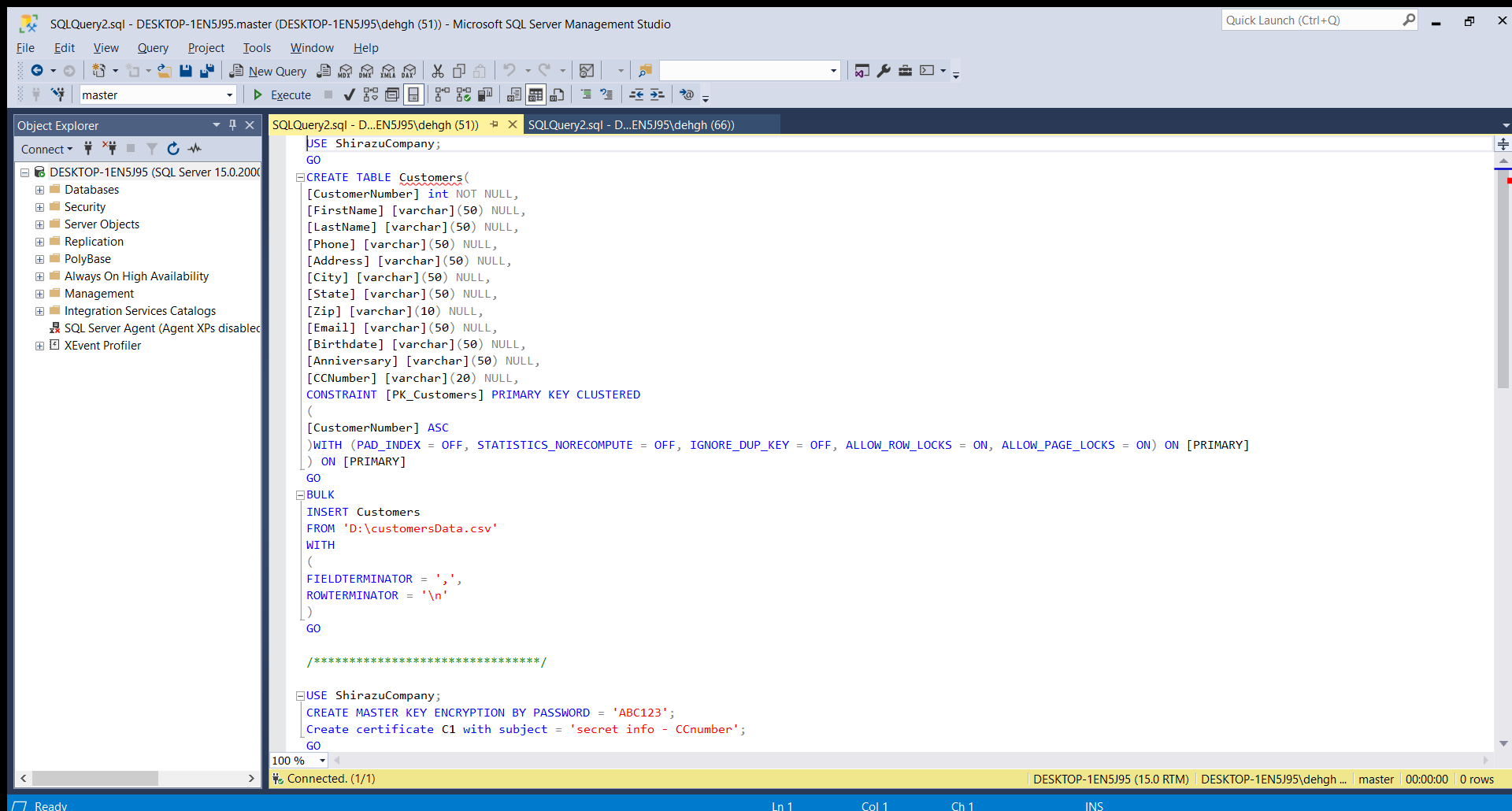
**Abstract:**

Data security is a critical task for any organization, especially if you store customer personal data such as Customer contact number, email address, social security number, bank and credit card numbers. Our main goal is to protect unauthorized access to data within and outside the organization. In this exercise gives an overview of column level SQL Server encryption.

**Environment set up:**

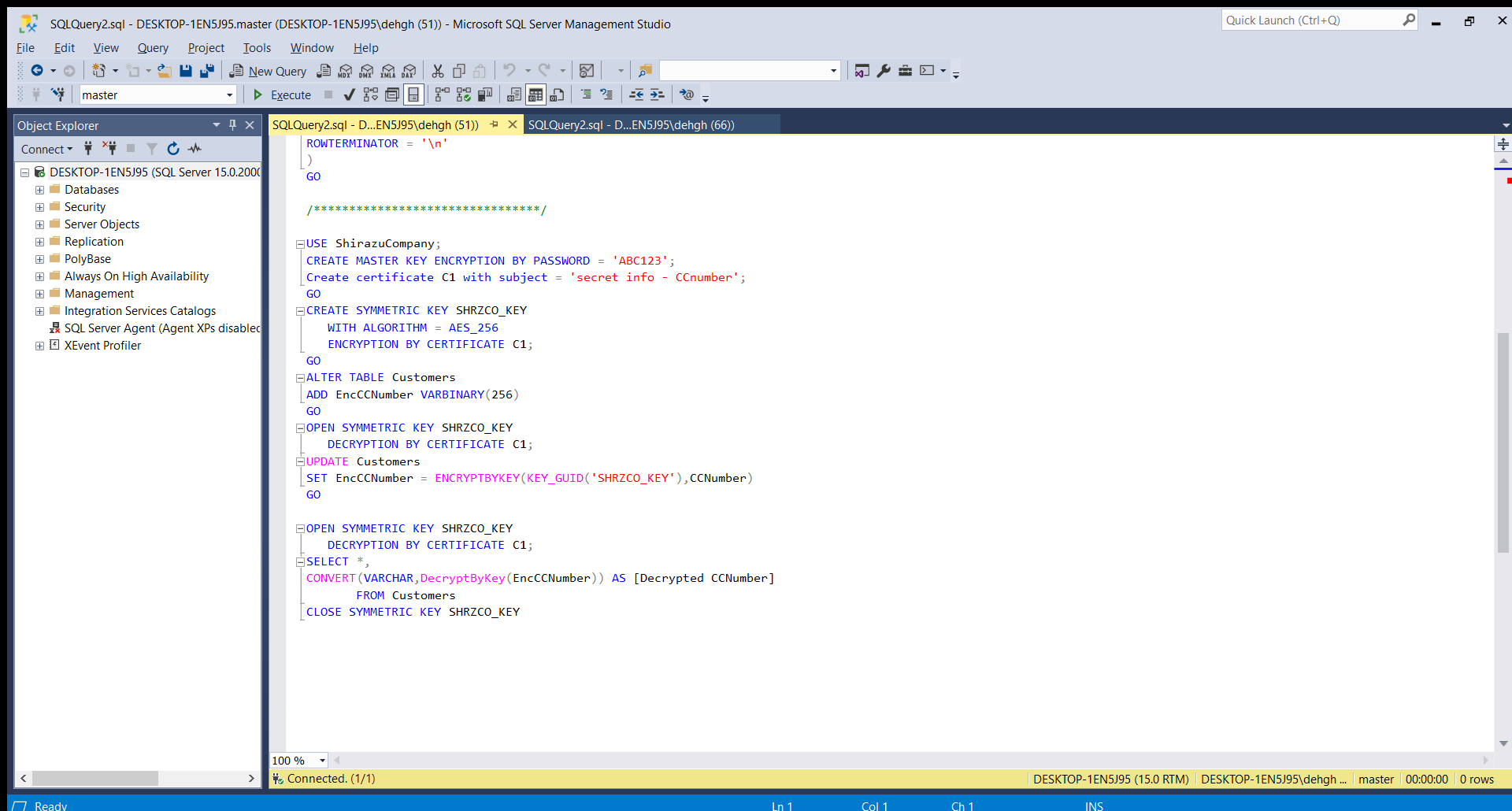
First create a new database as ShirazuCompany then create Customers table. For insert CustomerInfo table use Bulk , Bulk insert is a process or method provided by a database management system to load multiple rows of data into a database table, also we directed attached file .



We use the following steps for column level encryption:

1. Create a database master key
2. Create certificate for SQL Server
3. Configure a symmetric key for encryption
4. Encrypt the column data
5. Add new file
6. Query and verify the encryption



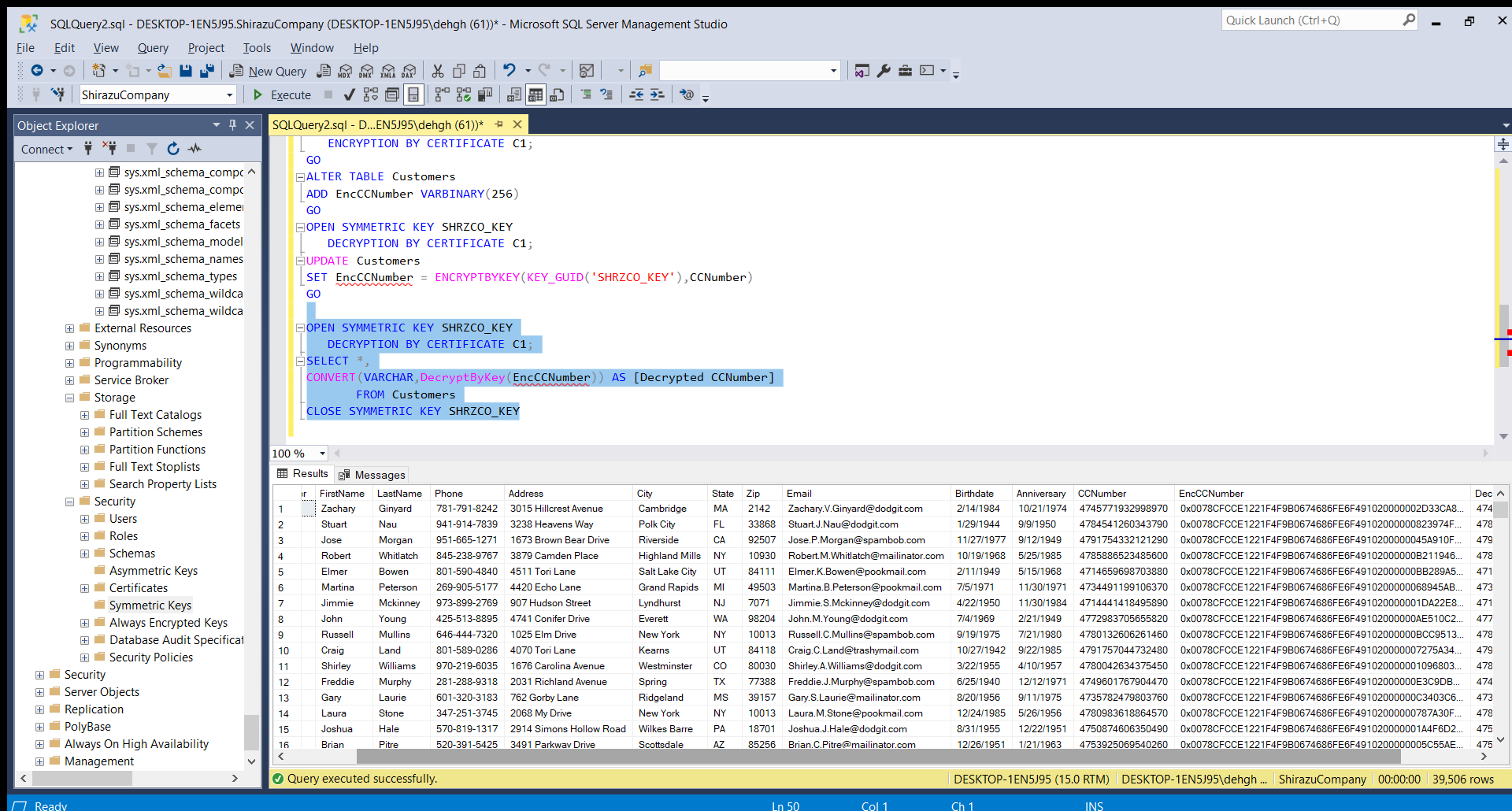
verify the encryption and also decrypted

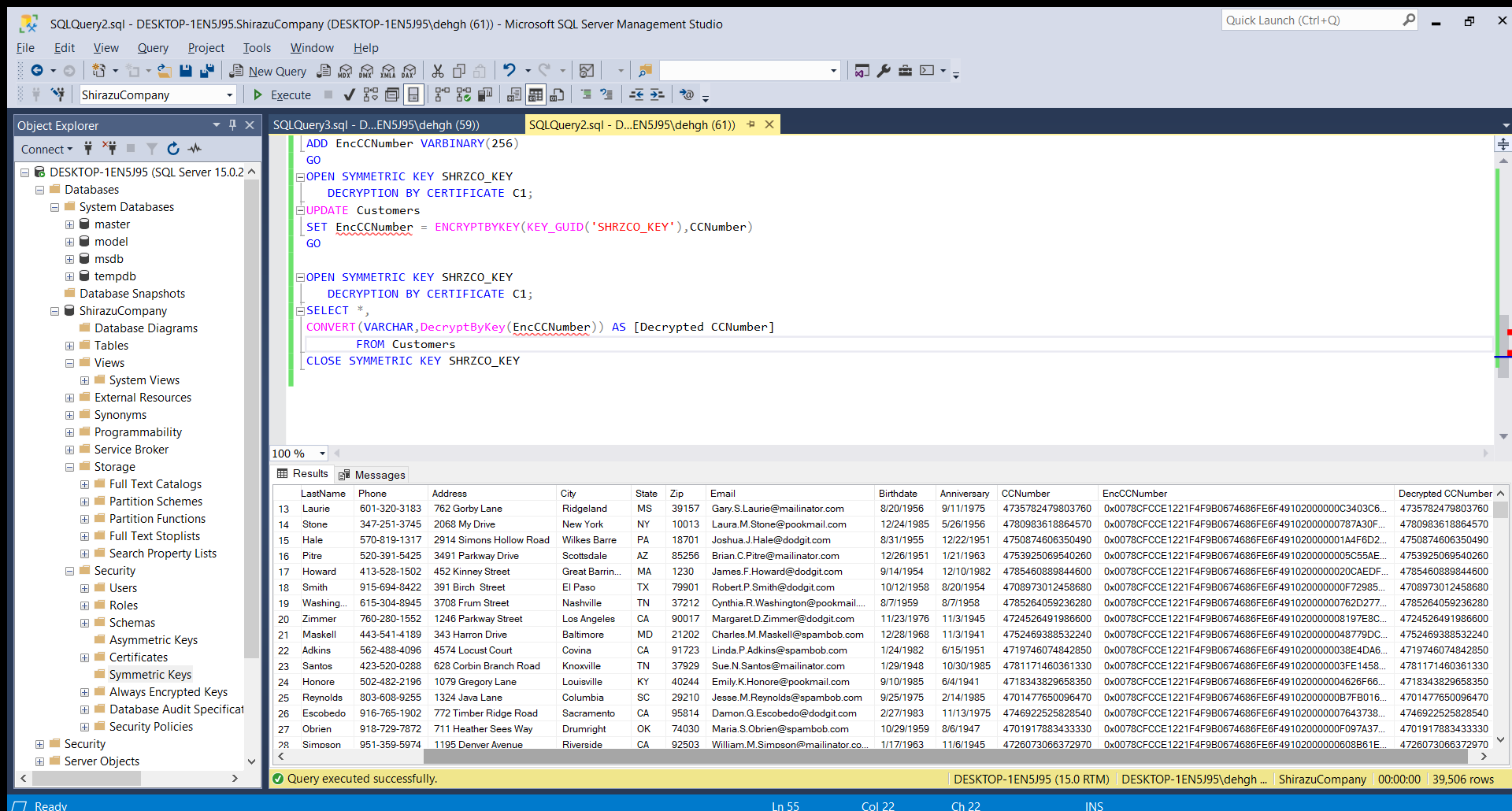
Create symmetric key with use of AES algorithm

Add new field and populate it with CCNumber column

Create master key and certificate

**What we see as a result :**

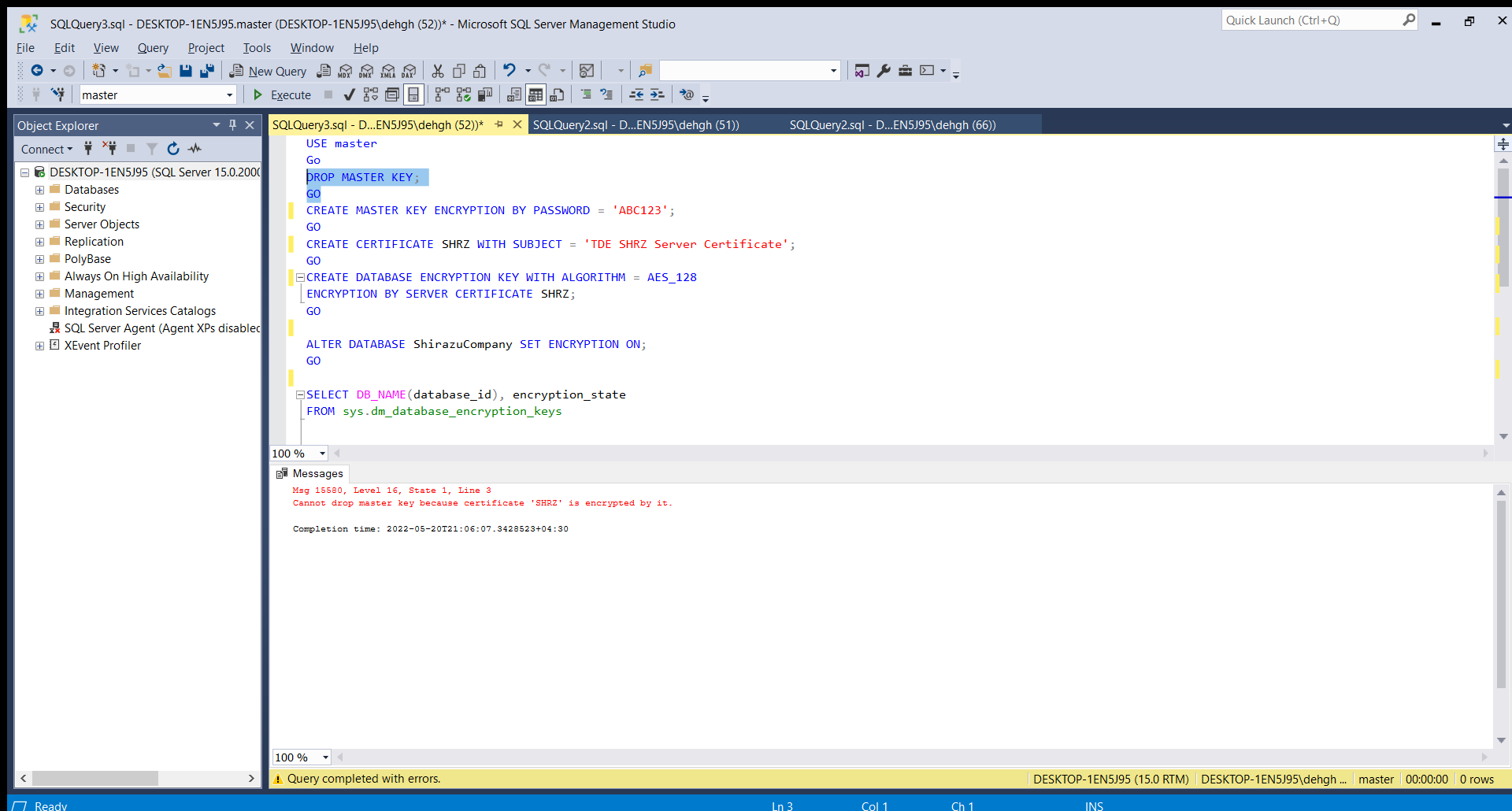
****



**TDE:**

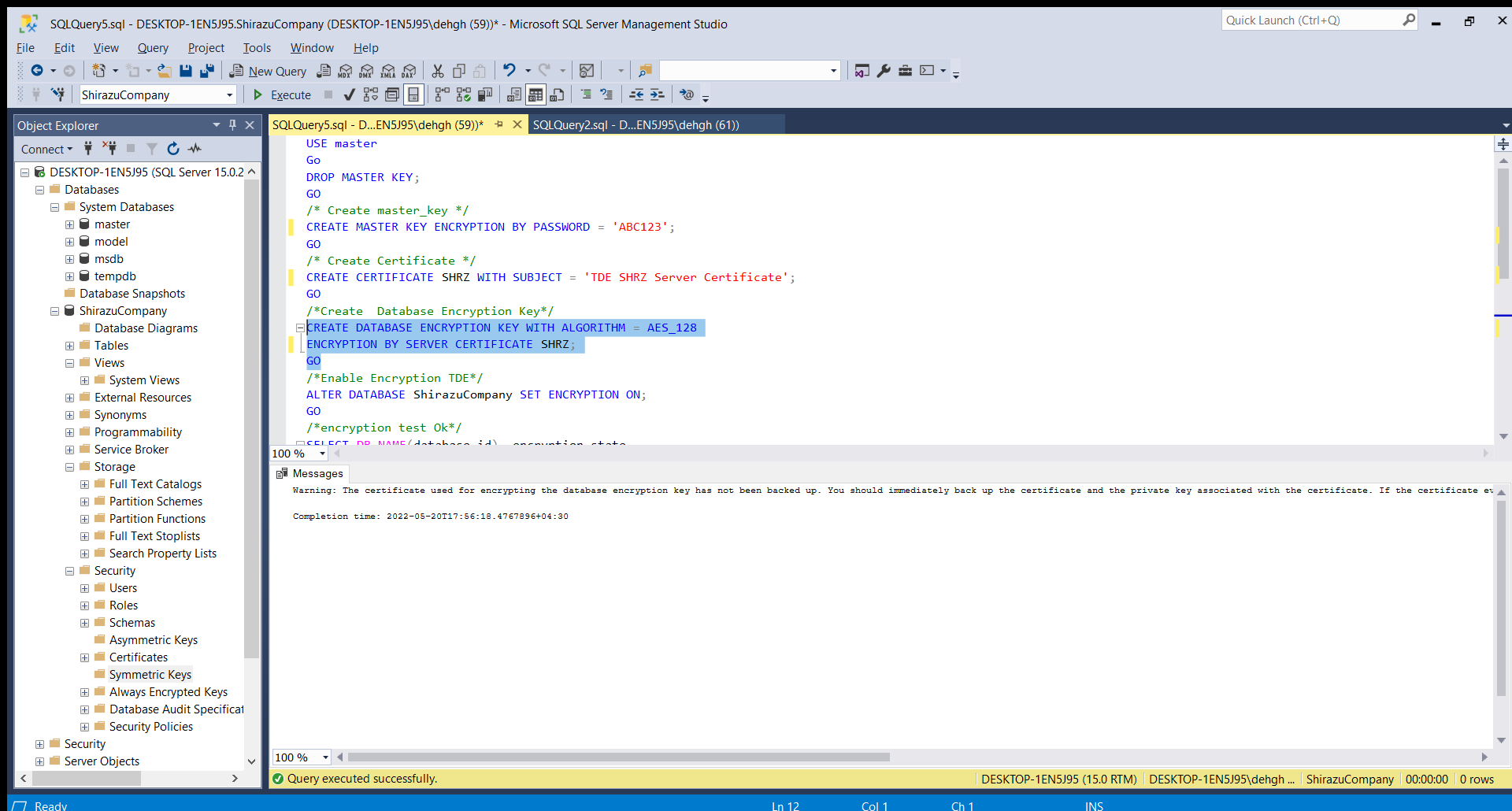
Here we should enable the TDE for shirazucompany database with the use of AES-128.

First step is drop master key but as show in the picture see no one to remove

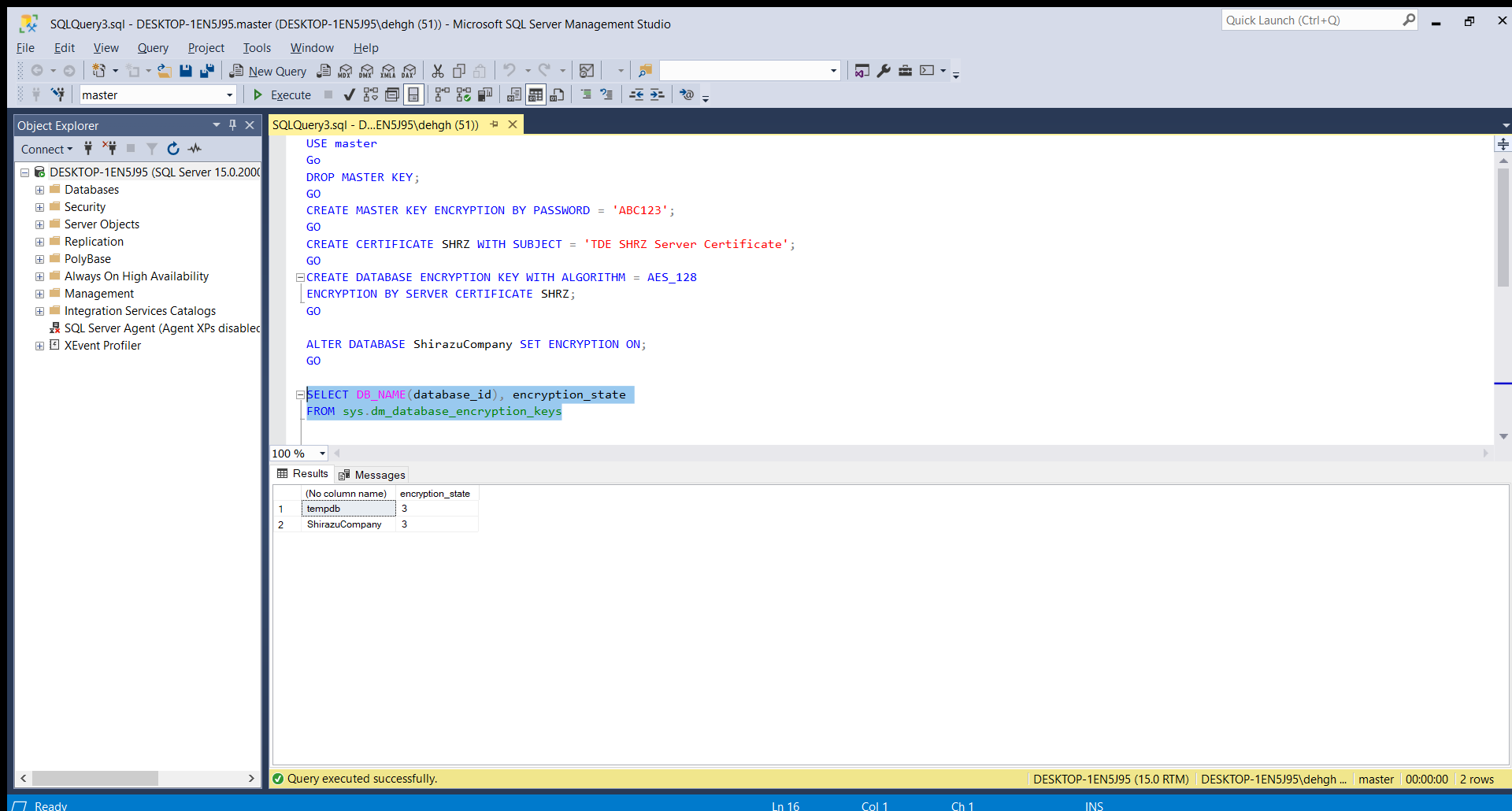


Then we create master key in encrypted mode , make a certificate for it.

Now we have to create database encryption key as we add algorithm type when execute this part it show a warning as below :



That should immediately back up the certificated and the private key associated with the certificate. Now the finial result is here:



**Conclusion :**

we explored column level SQL Server encryption using the symmetric key. We can use the same key for encrypting other table columns as well , should explore the encryption and decryption mechanism; however, we should consider the requirements first and then consider the appropriate encryption mechanism as per we need.