😊 Azure SQL – Data Masking

What is Data Masking?

Data Masking in Azure SQL Database is a **security feature** that helps protect sensitive data by **obscuring it from unauthorized users** while allowing authorized users to access the original data. Azure SQL Database provides **Dynamic Data Masking (DDM)**, which applies real-time obfuscation without altering the actual data in storage.

For example, a masked credit card number might appear as **XXXX-XXXX-XXXX-1234** instead of the full number.

Benefits of Data Masking

1. Protects Sensitive Data

- Hides personal and confidential data from unauthorized access.

2. No Changes to the Database Structure

- Works at the query level without modifying the actual data.

3. Easy to Implement

- Configured directly in **Azure SQL Database** with simple rules.

4. Reduces Security Risks

- Prevents **accidental data exposure** to non-privileged users.

5. Compliance with Regulations

- Helps meet **GDPR, HIPAA, PCI DSS**, and other data privacy laws.

Use Cases of Data Masking

1. Protecting Personally Identifiable Information (PII)

- Masking names, email addresses, and contact numbers in customer databases.

2. Securing Financial Data

- Hiding credit card numbers, bank account details, and salaries from unauthorized users.

3. Healthcare Data Protection

- Masking patient records, medical history, and insurance details.

4. Development & Testing Environments

- Allowing developers to work with realistic but **obfuscated** data.

5. Preventing Insider Threats

- Ensuring that employees only see **masked sensitive data** based on their roles.

Conclusion

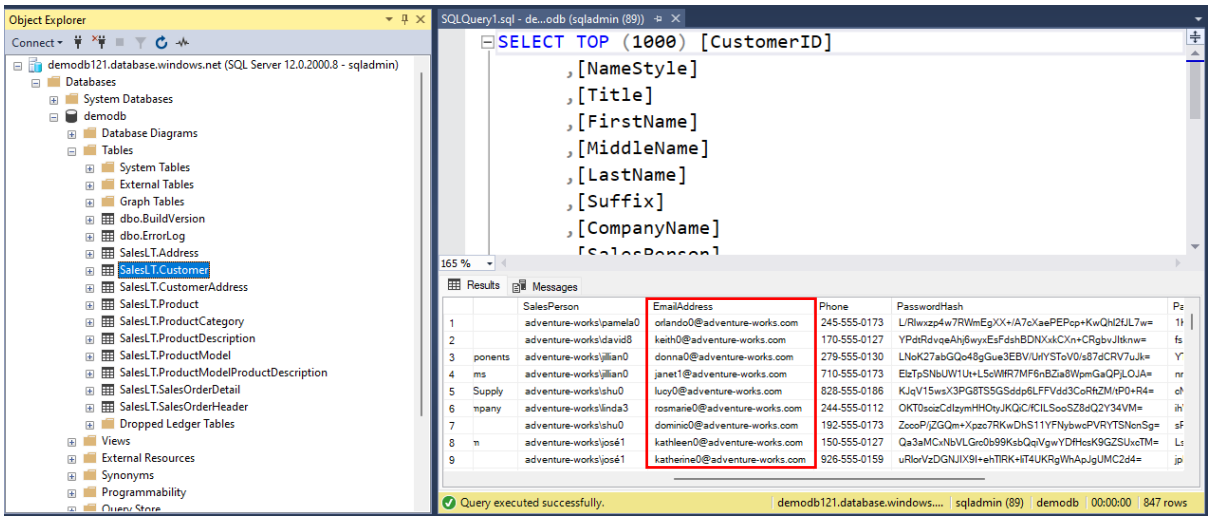
Azure SQL **Data Masking** is an **efficient and simple way to protect sensitive data** without modifying its actual storage. It helps maintain **data security, privacy, and regulatory compliance** while allowing controlled access to necessary information.

In this lab, we enable Dynamic Data Masking (DDM) in Azure SQL Database to obscure sensitive data from unauthorized users while keeping it accessible to admins. First, log in to SQL Server Management Studio (SSMS) and identify the target column (e.g., email addresses). Then, configure data masking via the Azure Portal by adding a masking rule to the column. Next, create a new user with restricted access and log in as them. The masked data is visible for this user but not for admins.

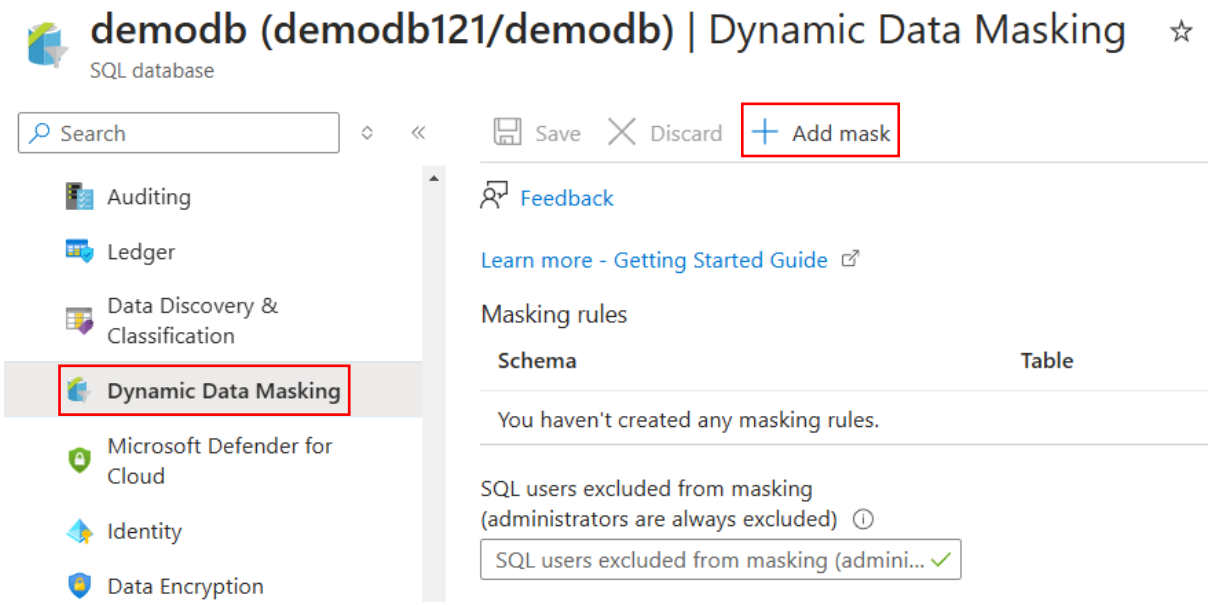
End Goal: Implement data masking to enhance security, prevent unauthorized data access, and comply with data privacy regulations.

😄 To begin with the Lab

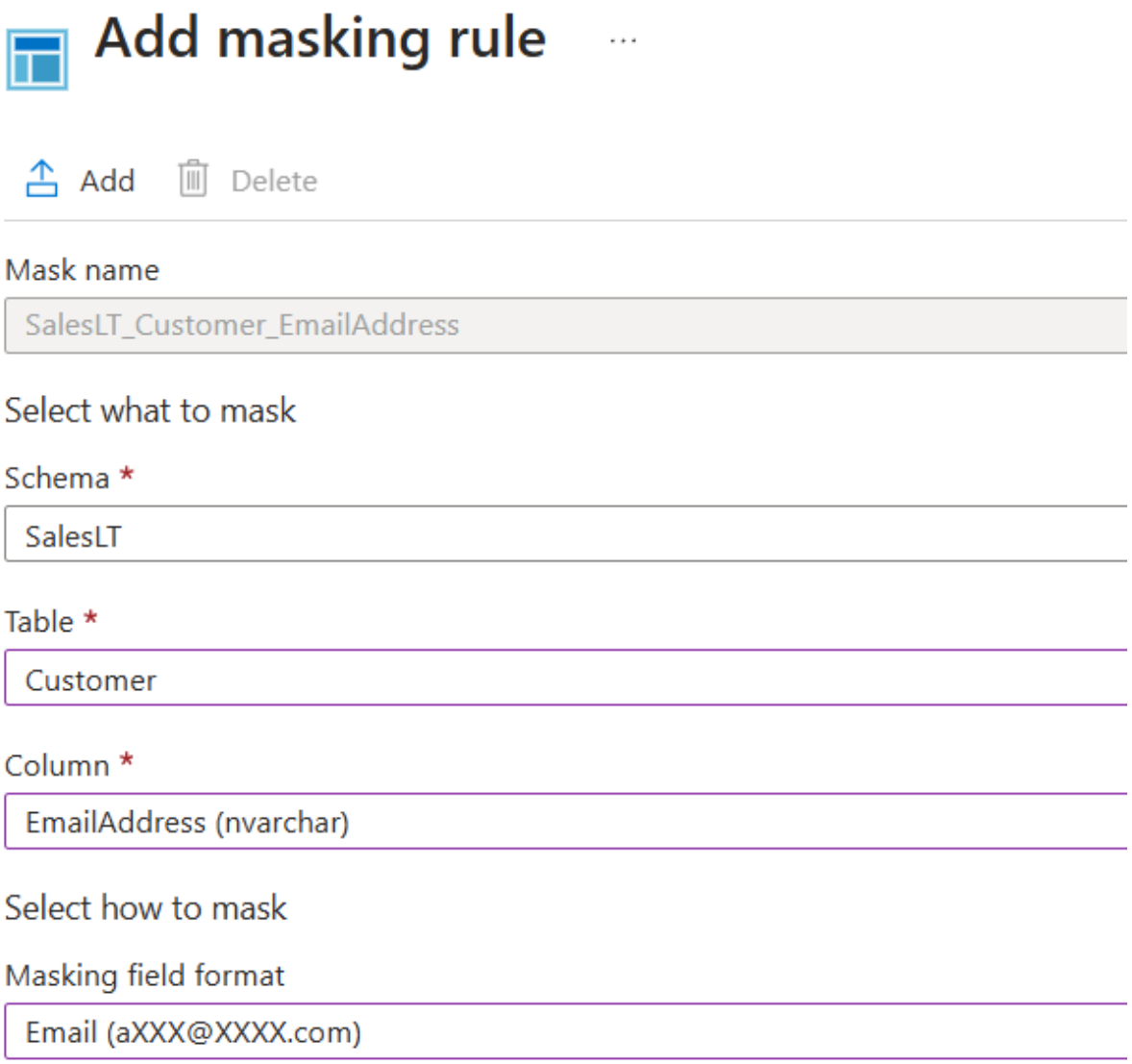
1. In this lab, we will see the data masking feature in Azure SQL Database.
2. The prerequisite for this lab is that you should have the Azure SQL Database running in your Azure portal with sample data loaded on it.
3. So, log in to your SQL Server using SQL Server Management Studio. Here you will see a table with the name Customer.
4. Inside this table you will see a column for email addresses and let's say we want to mask the information for the email address.



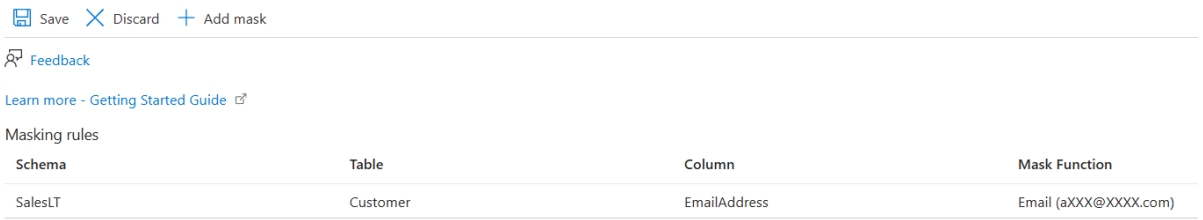
5. For that first we will go to our SQL Database on the Portal. Here we have a feature called dynamic data masking. We can choose to add a mask.



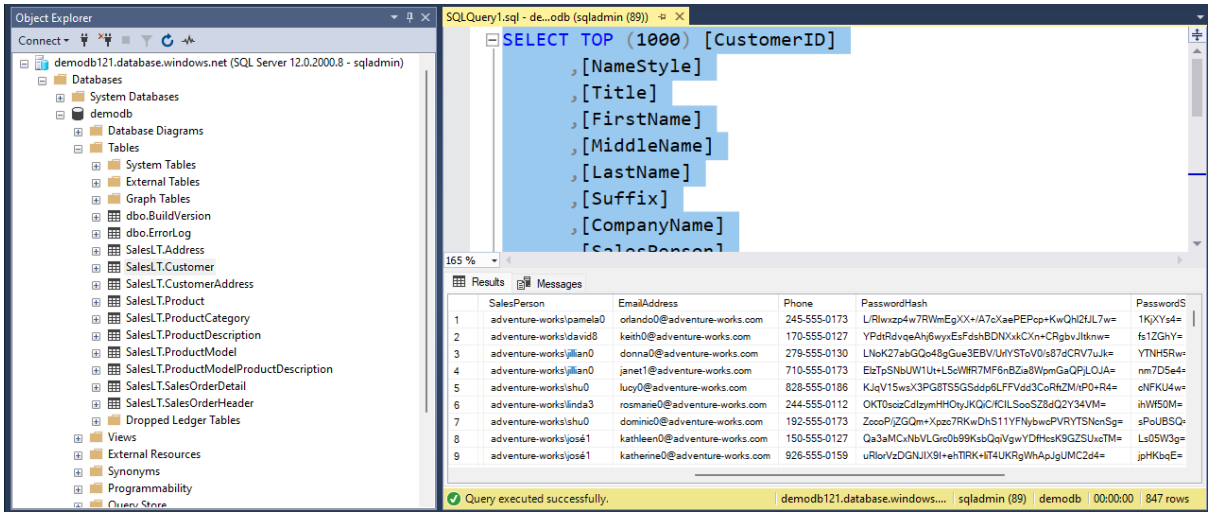
6. In the add masking rule you can choose the Customer table and then choose the Email column. After that, you have different options to choose the masking field format. Click on Add.



7. Once it is added click on the save button as well.



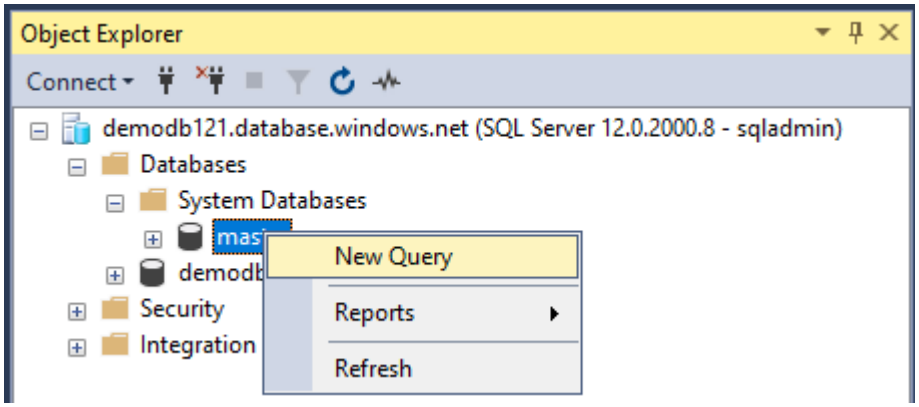
8. Now if I come back to SQL Server Management Studio and execute the query again, we will see that the masking does not happen here.
9. It is because we are logged in as the admin user and the masking does not affect your admin level users.



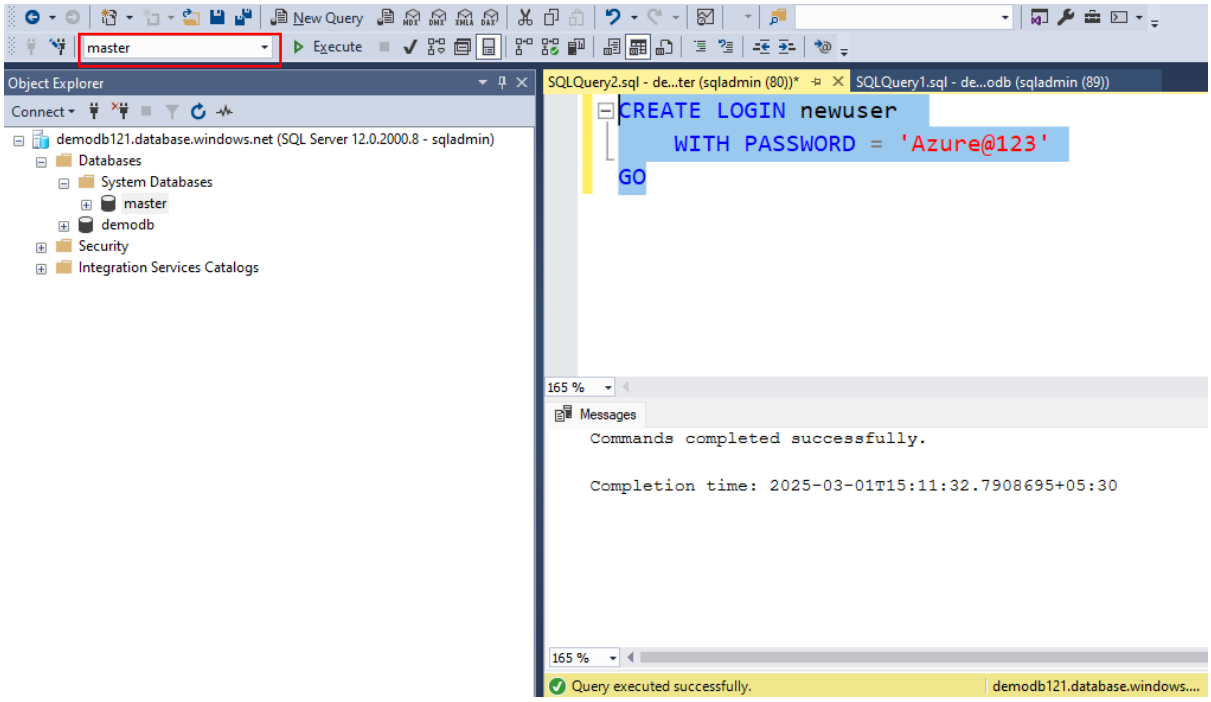
10. For this we are going to create a new user using the below commands and add a role as well to read the data.

```
CREATE LOGIN newuser
    WITH PASSWORD = 'Azure@123'
GO


CREATE USER newuser
    FOR LOGIN newuser
    WITH DEFAULT_SCHEMA = SalesLT


EXEC sp_addrolemember 'db_datareader', 'newuser';

DATABASE=demodb
```

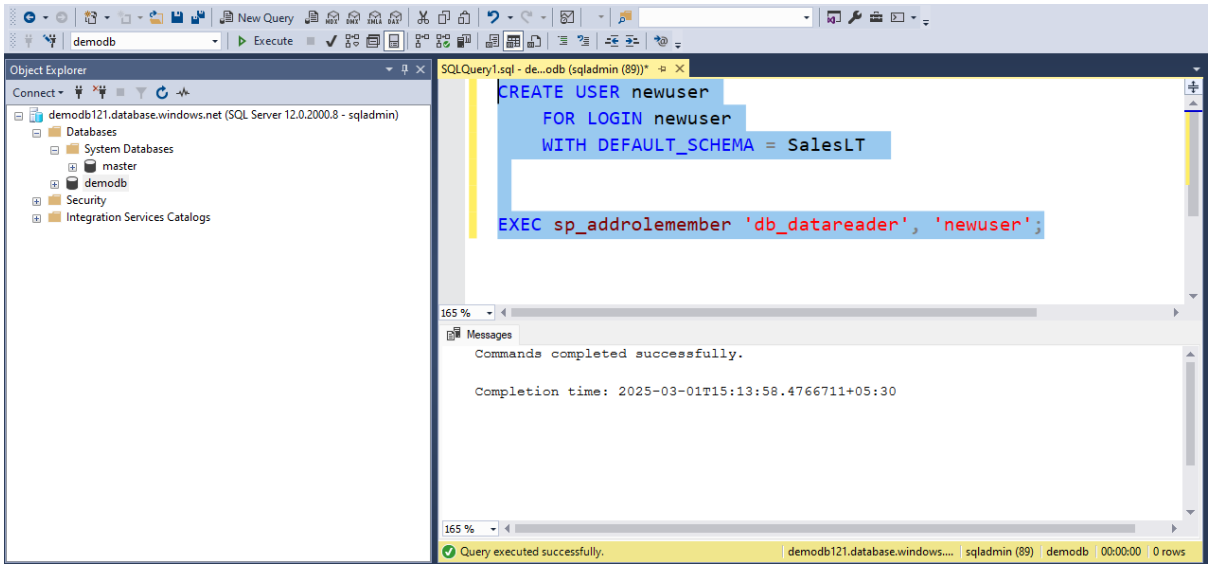
11. First you need to expand system databases, then right click on the master database and choose to create a new query.



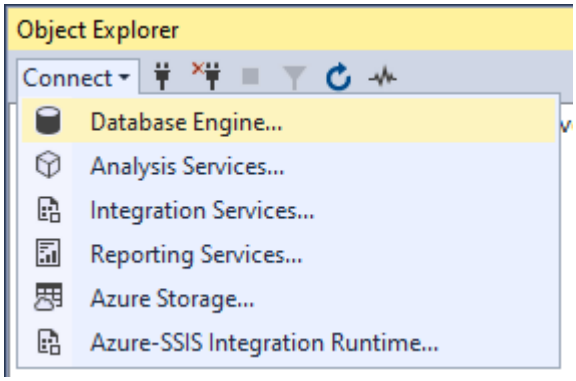
12. Then a new query page will be opened on the canvas using the same command shown below you have to create a new login for the new user along with the password as shown in the snapshot.
13. Also you can see that we are in the master database as highlighted.



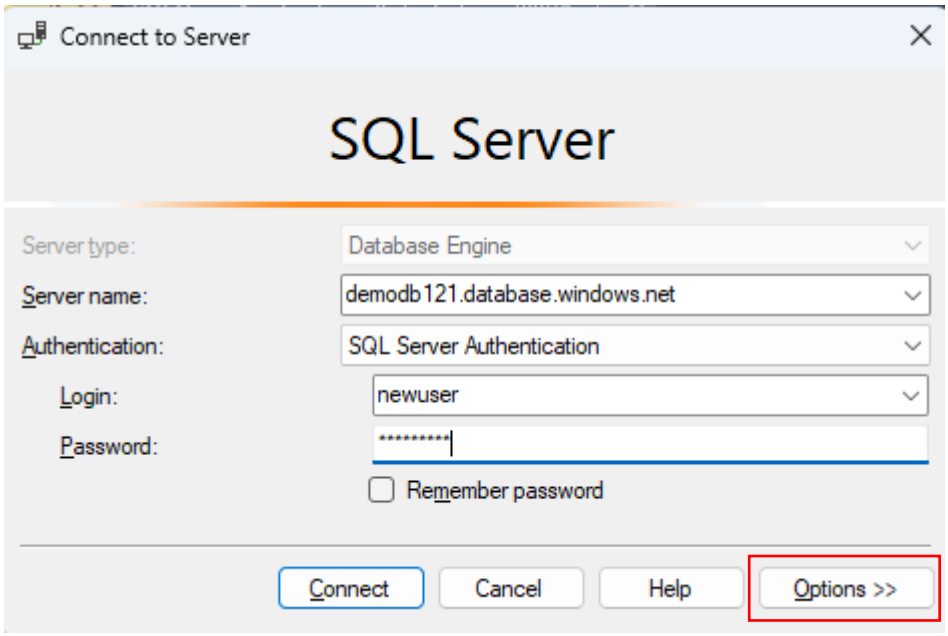
14. Then you need to change your database to demo DB and fire up the remaining commands.



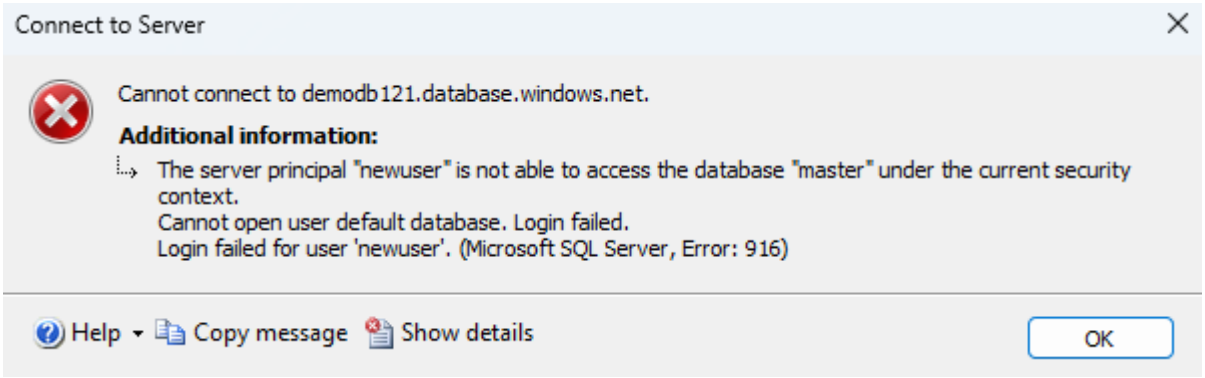
15. Now from the data explorer, click on connect then choose database engine to login as the new user we just created.



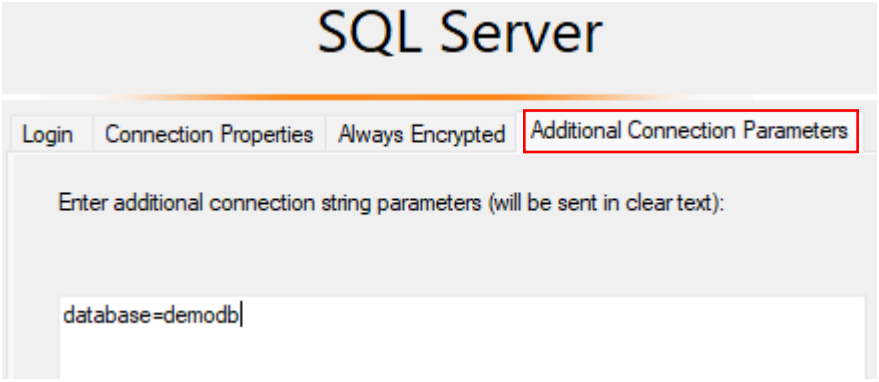
16. Here you just need to give the login name and password, then click on connect.



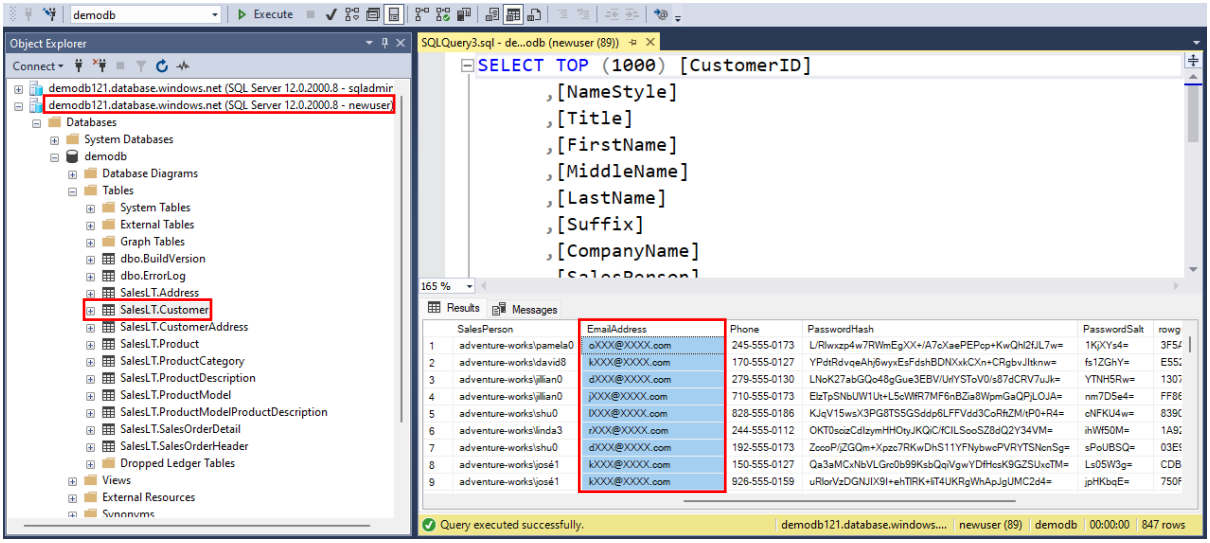
17. So, you will see this error when you click on Connect. To resolve this error, you have to click on the options button as shown above in the snapshot.



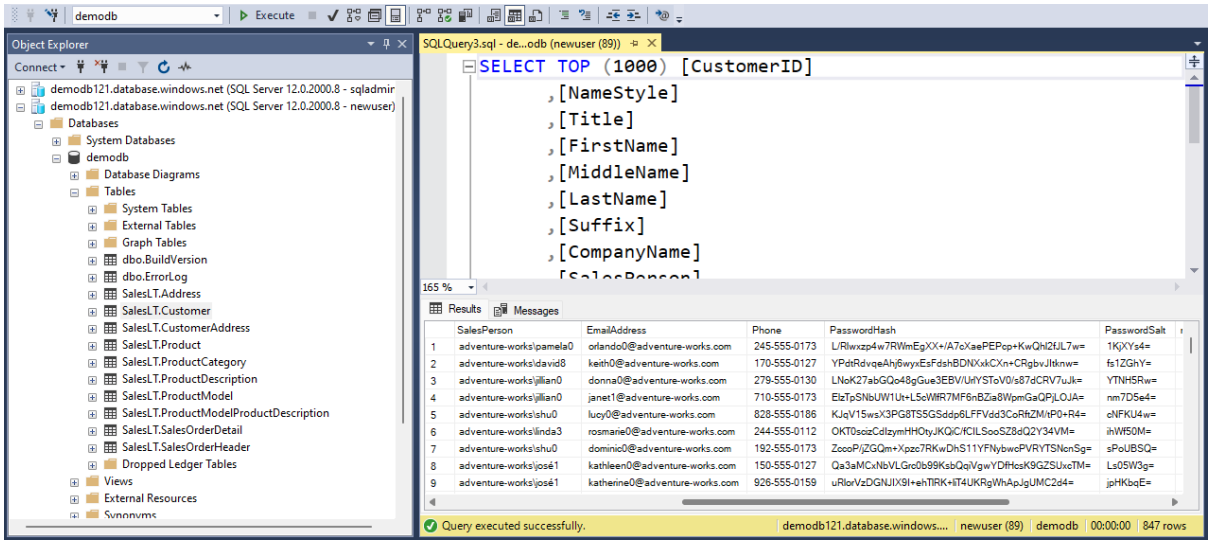
18. Then move to Additional connection parameters and here you have to define your database. Click on Connect.



19. Here in the snapshot you can see that we are connected with the new user and when we displayed the data from customer table in the email column we can see the masking.



20. Now if we delete the masking from the portal and save the changes then execute the query again. We can see the email addresses.


demodb
Execute
Object Explorer
Connect
demodb121.database.windows.net (SQL Server 12.0.2000.8 - sqladmir
demodb121.database.windows.net (SQL Server 12.0.2000.8 - newuser)
Databases
System Databases
demodb
Database Diagrams
Tables
System Tables
External Tables
Graph Tables
dbo.BuildVersion
dbo.ErrorLog
SalesLT.Address
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProductDescription
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Dropped Ledger Tables
Views
External Resources
Synonyms
SQLQuery3.sql - de...odb (newuser (89))
SELECT TOP (1000) [CustomerID]
,[NameStyle]
,[Title]
,[FirstName]
,[MiddleName]
,[LastName]
,[Suffix]
,[CompanyName]
165 %
Results
Messages
SalesPerson
EmailAddress
Phone
PasswordHash
PasswordSalt
1 adventure-works\pamela0 orlando0@adventure-works.com 245-555-0173 L/Rlwxzp4w7RWmEgXX+/A7cXaePEPcp+KwQhl2fJL7w= 1KjXYs4=
2 adventure-works\david8 keith0@adventure-works.com 170-555-0127 YPdtRdvqeAhj6wyxEsFdshBDNXxkCXn+CRgbvJltknw= fs1ZGhY=
3 adventure-works\jillian0 donna0@adventure-works.com 279-555-0130 LNoK27abGQo48gGue3EBV/UrlYSToV0/s87dCRV7uJk= YTNHSRw=
4 adventure-works\jillian0 janet1@adventure-works.com 710-555-0173 ElzTpSNbUW1Ut+L5cWfR7MF6nBZia8WpmGaQPJLOJA= nm7D5e4=
5 adventure-works\shu0 lucy0@adventure-works.com 828-555-0186 KJqV15wsX3PG8TS5GSddp6LFFVdd3CoRftZM/tP0+R4= cNFKU4w=
6 adventure-works\linda3 rosmarie0@adventure-works.com 244-555-0112 OKT0sciaCdlzymHHOtyJKQiC/fCILSooSZ8dQ2Y34VM= ihWf50M=
7 adventure-works\shu0 dominic0@adventure-works.com 192-555-0173 ZocoP/jZGQm+Xpzo7RKwDhS11YFNybwcPVRYTSNcnSg= sPoUBSQ=
8 adventure-works\josé1 kathleen0@adventure-works.com 150-555-0127 Qa3aMCxNbVLGrc0b99KsbQqiVgwYDfHosK9GZSUxcTM= Ls05W3g=
9 adventure-works\josé1 katherine0@adventure-works.com 926-555-0159 uRlorVzDGNJIX9I+ehTlRK+lIT4UKRgWhApJgUMC2d4= jpHKbqE=
Query executed successfully.
demodb121.database.windows....
newuser (89)
demodb
00:00:00
847 rows