

# IMPLEMENTING SECURITY FEATURES IN OS & DB

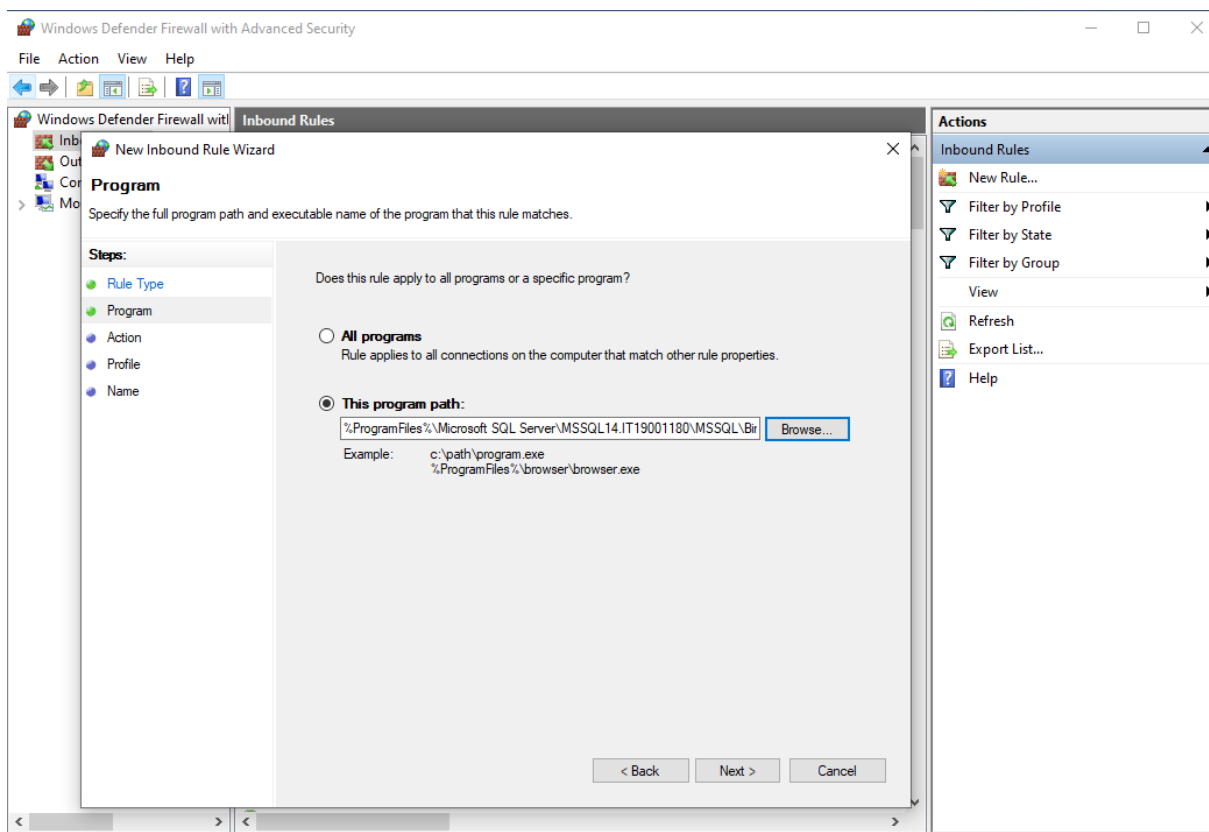
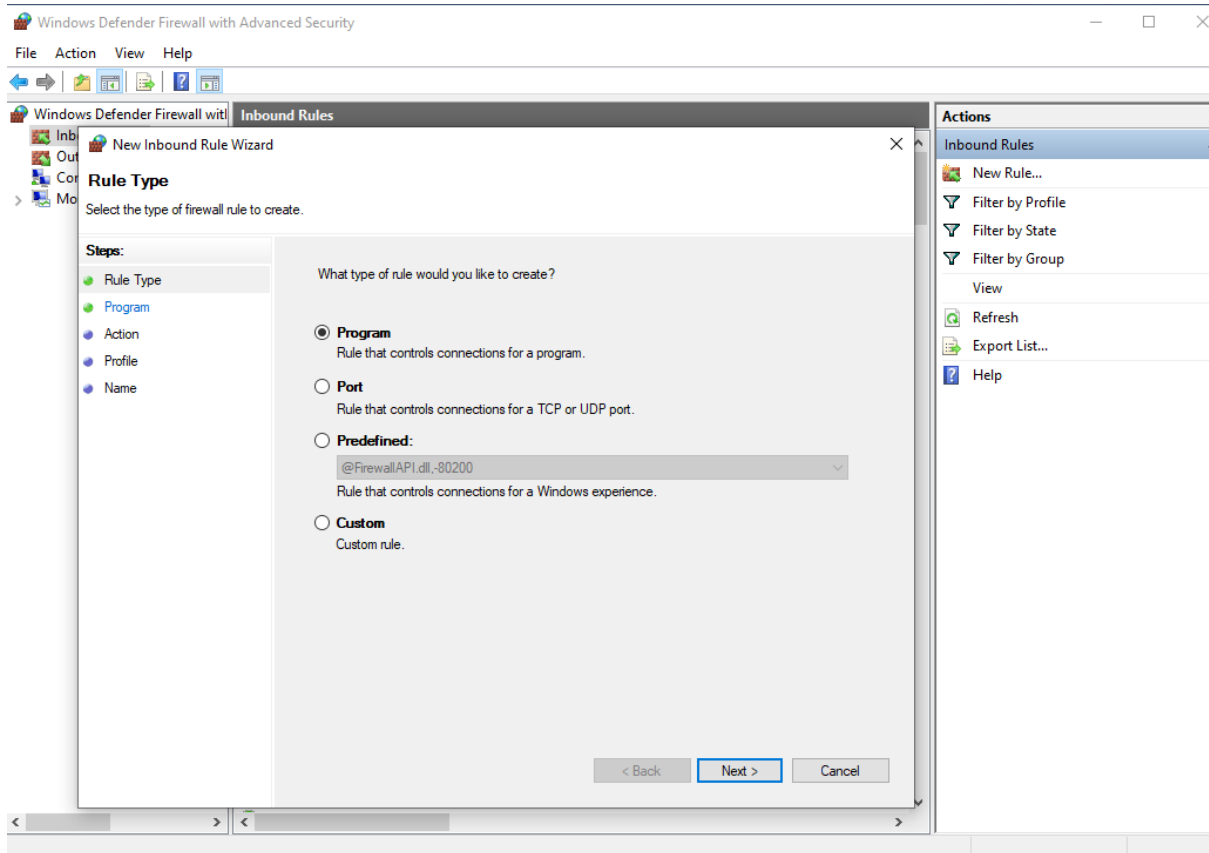
DOSS – IE3062

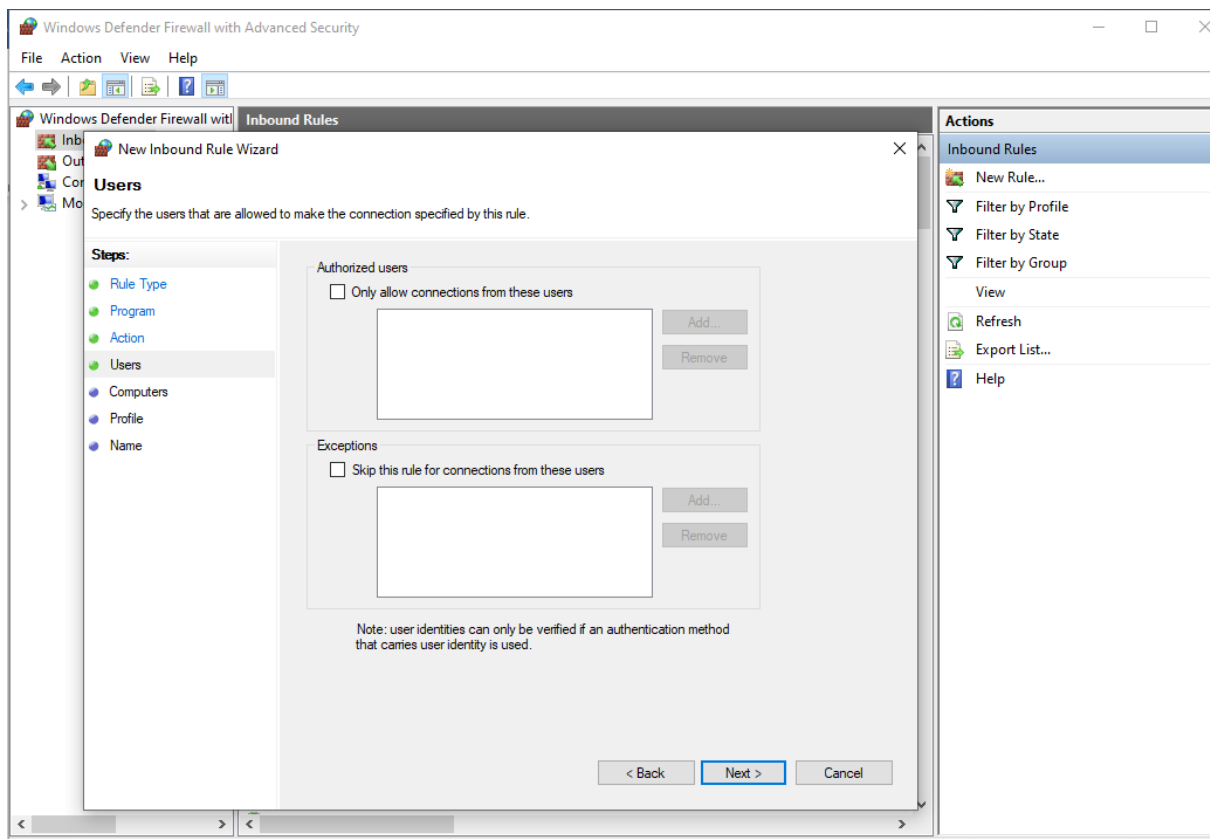
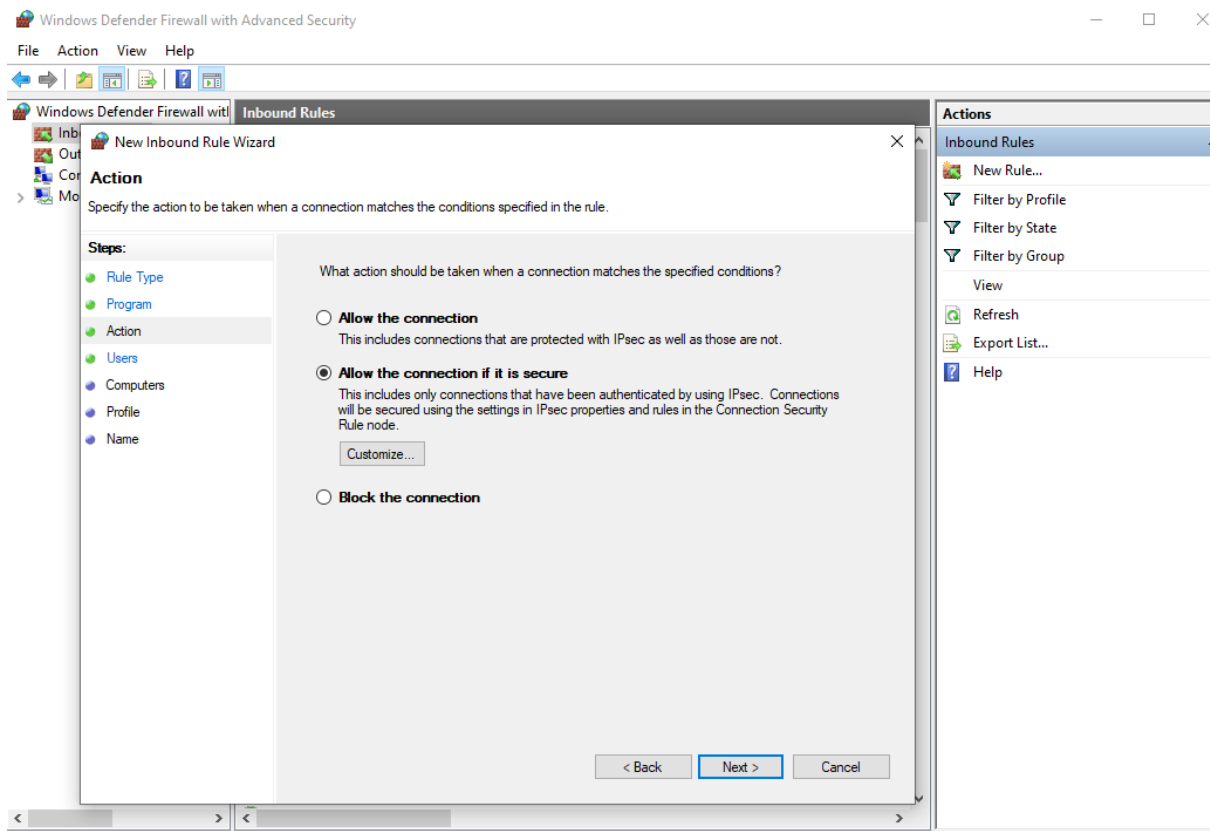
## Contents

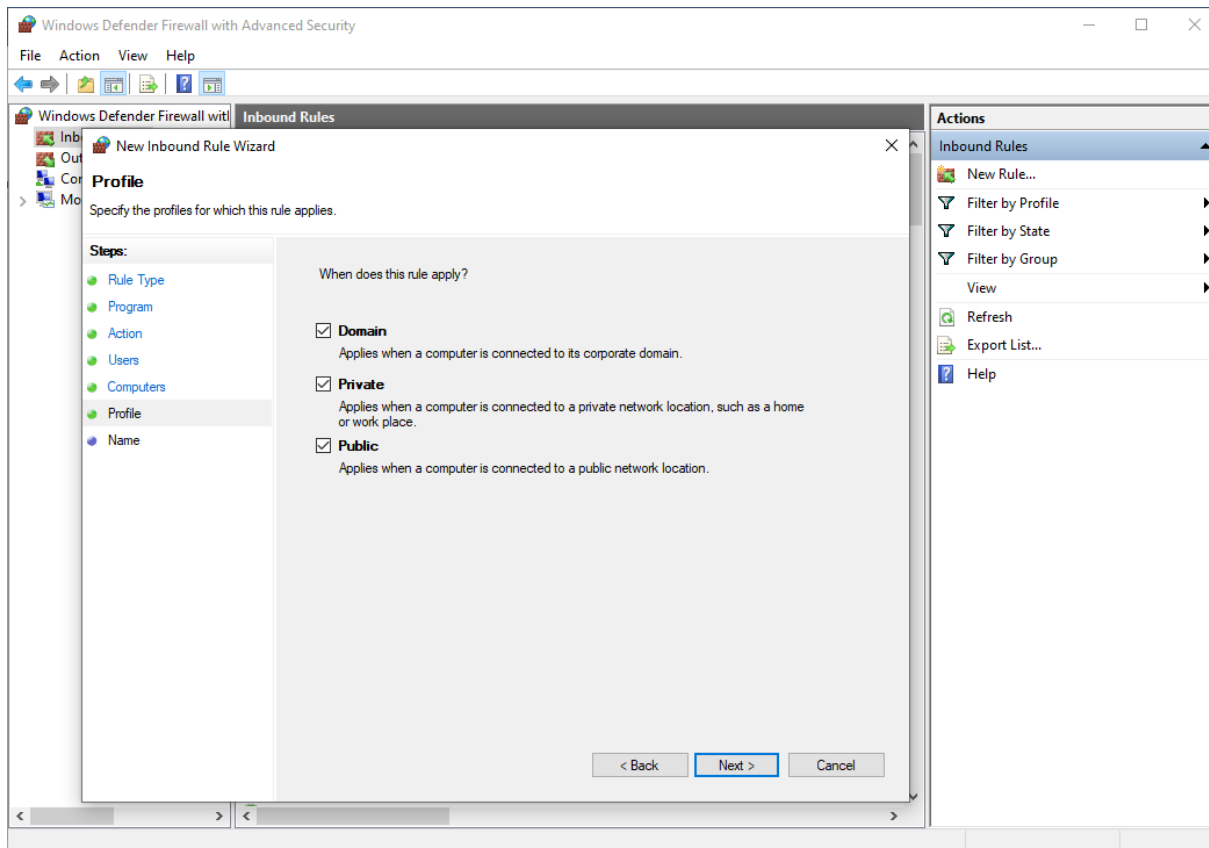
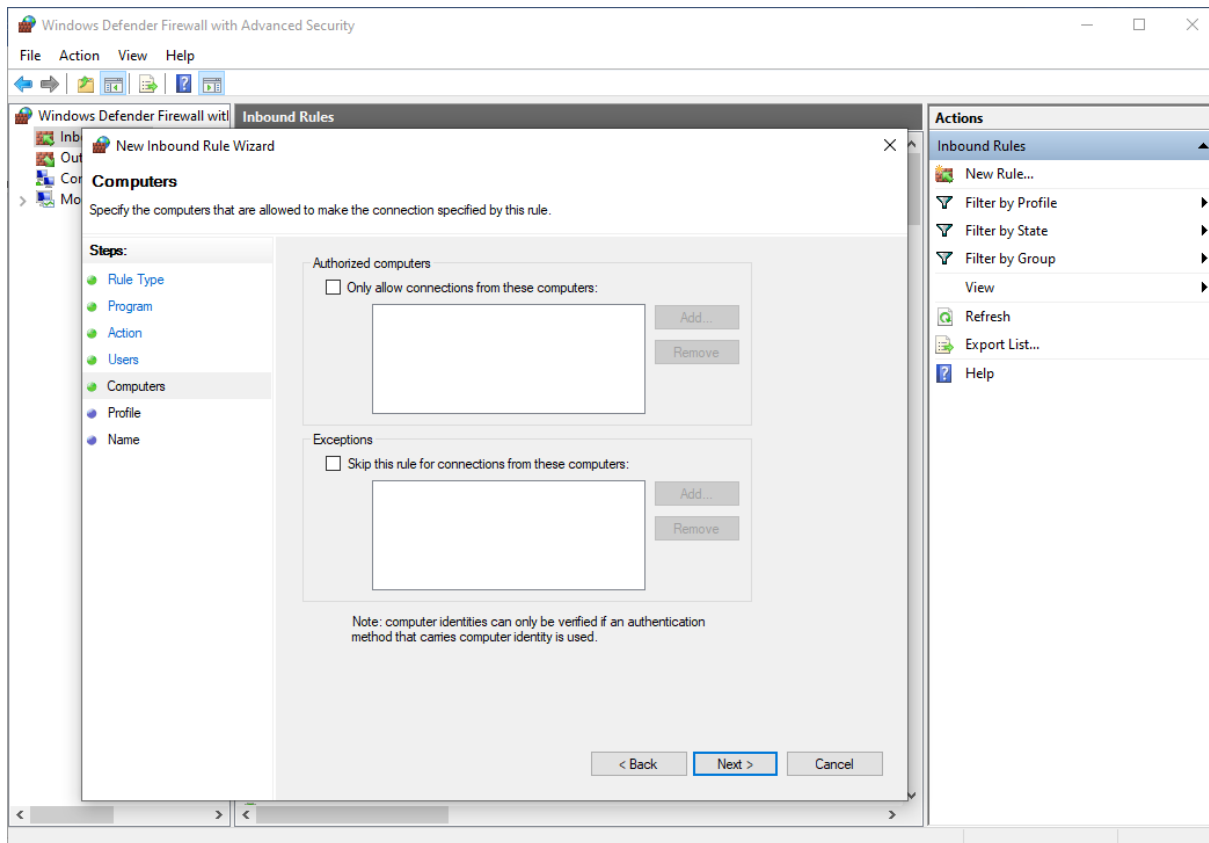
Firewall for DB Server.....	2
Inbound rule for SQL Server .....	2
Inbound rule for TCP ports .....	6
Configuring Windows firewall rules with PowerShell .....	8
Trying to log into the DB using the IP / Remote access to the DB.....	11
Database Software & Maintain individual login credentials for the people who access the workstation and to perform administrative tasks of the database .....	14
sa – System Administrator .....	14
Security_Admin .....	14
User_G .....	15
Maintain the log records of accessing to the database and maintain the minimum access privileges to the existing servers and applications & Turn on Auditing where technically possible for the database objects with protected data.....	16
Server level configuration settings.....	17
Grant minimal permissions that necessary for the people according to their job role in the database .	19
dbo – DB Owner .....	19
Guest .....	20
Permissions should be managed through roles or groups and not by direct grants to User IDs where possible .....	20
Security_Admin .....	20
User_G .....	22
Manage to use strong password and follow secure methods to preserve the stored passwords.....	25
Hashing .....	25
Encryption.....	28
Prevent from redundancy of the stored records of the database .....	29
Discuss how manage the implemented database backup and recovery .....	30

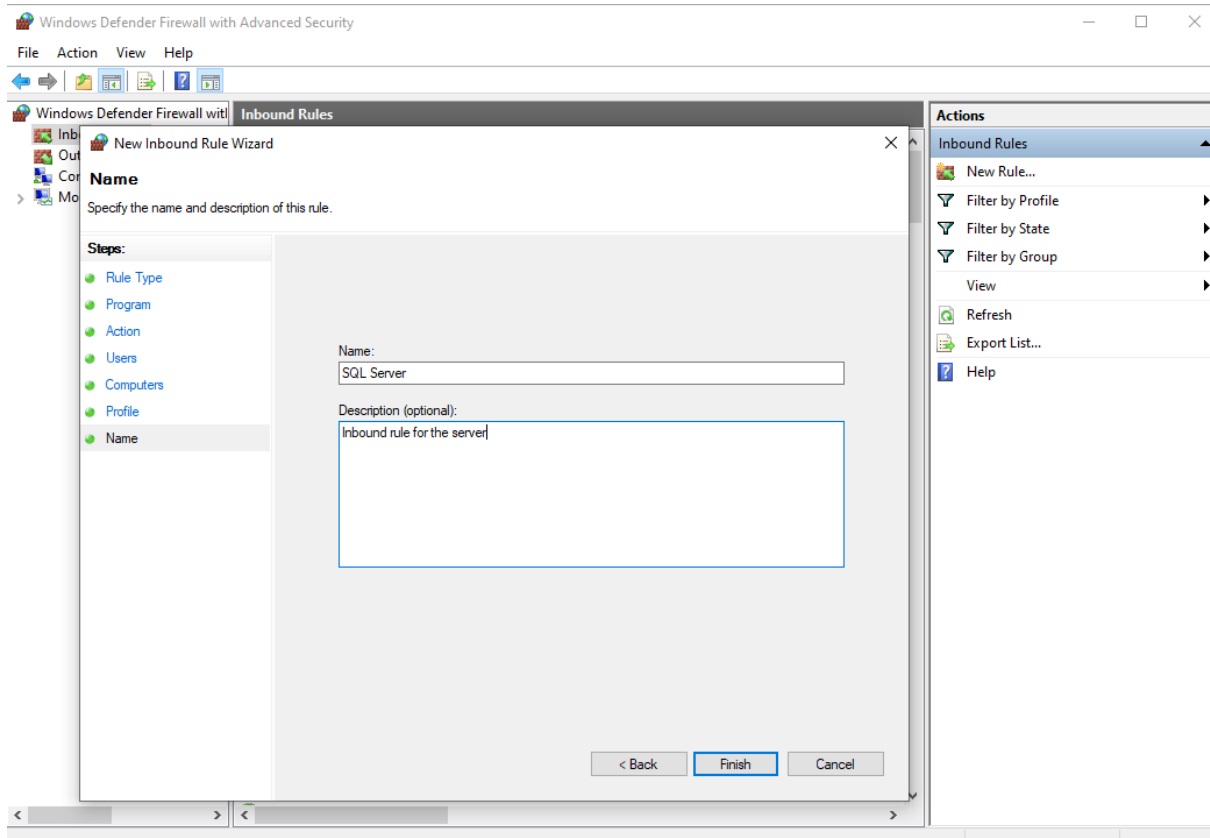
# Firewall for DB Server

## Inbound rule for SQL Server

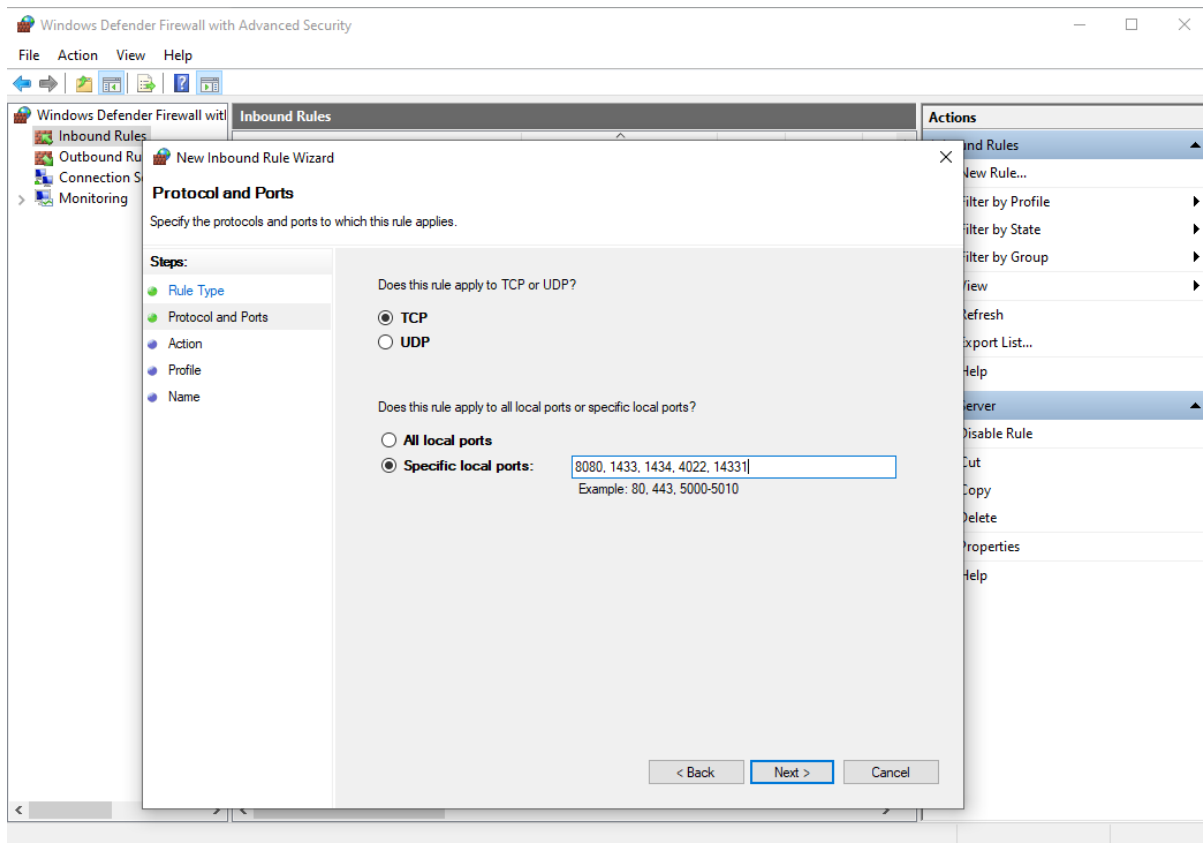
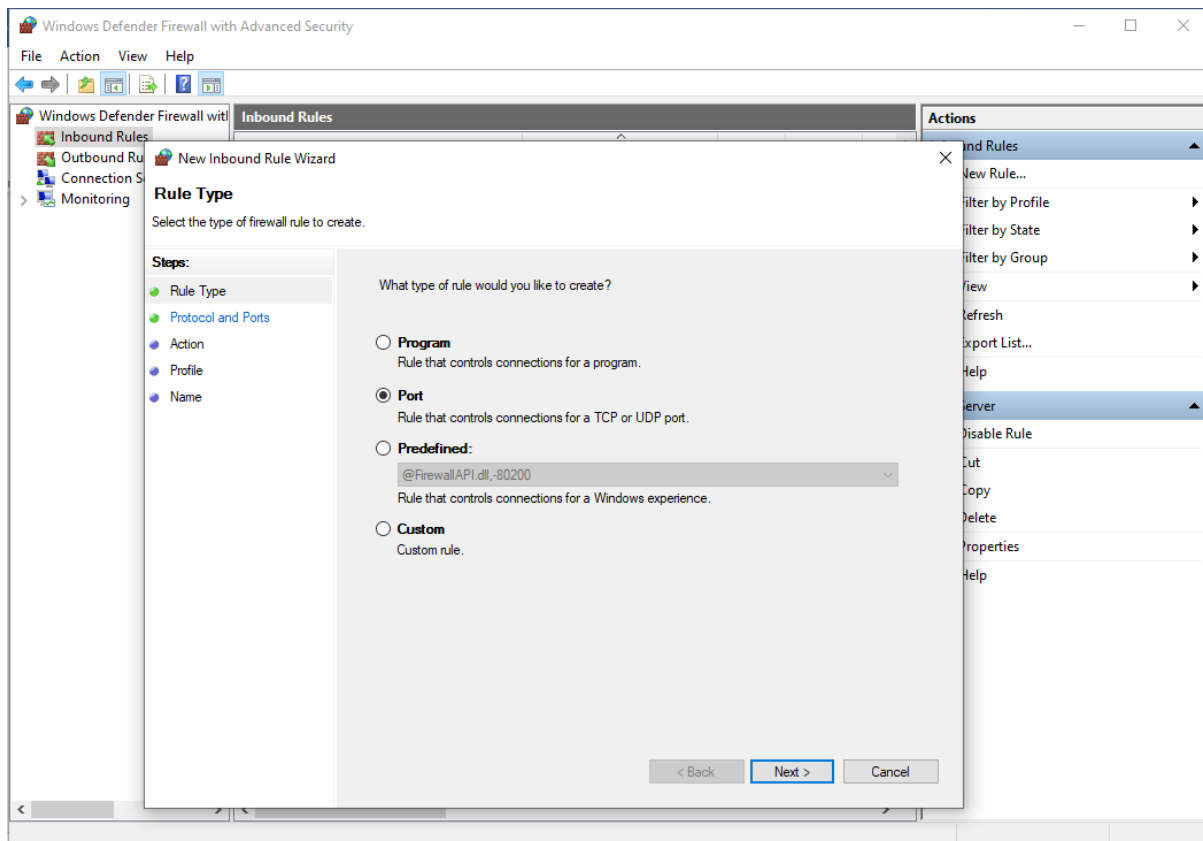


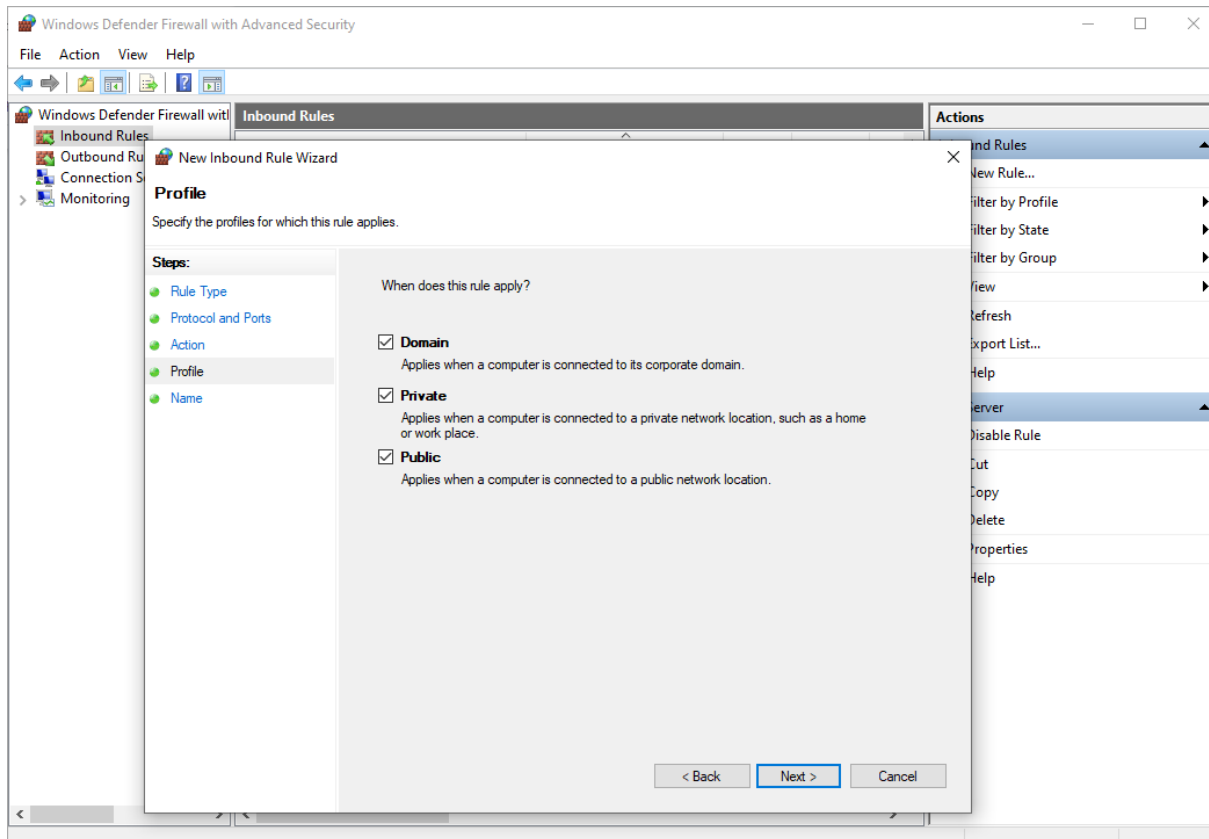
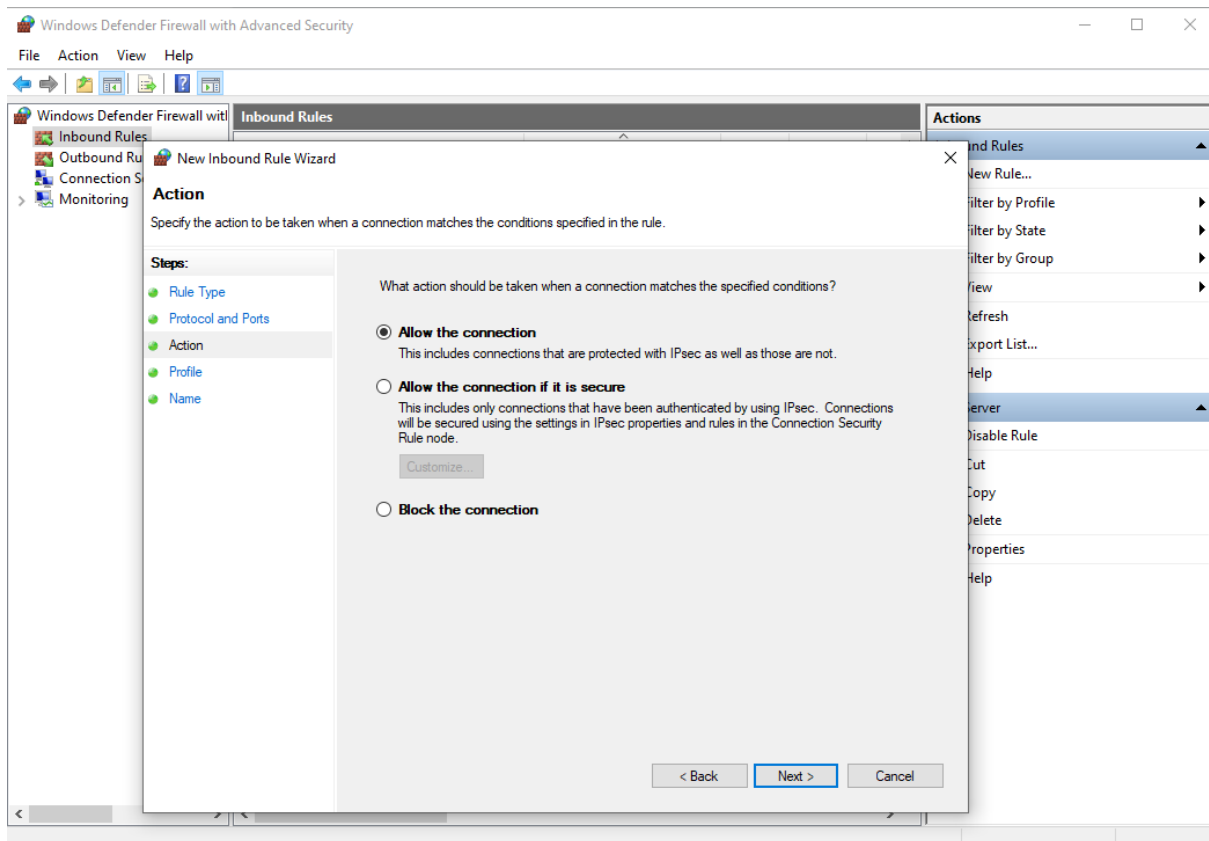




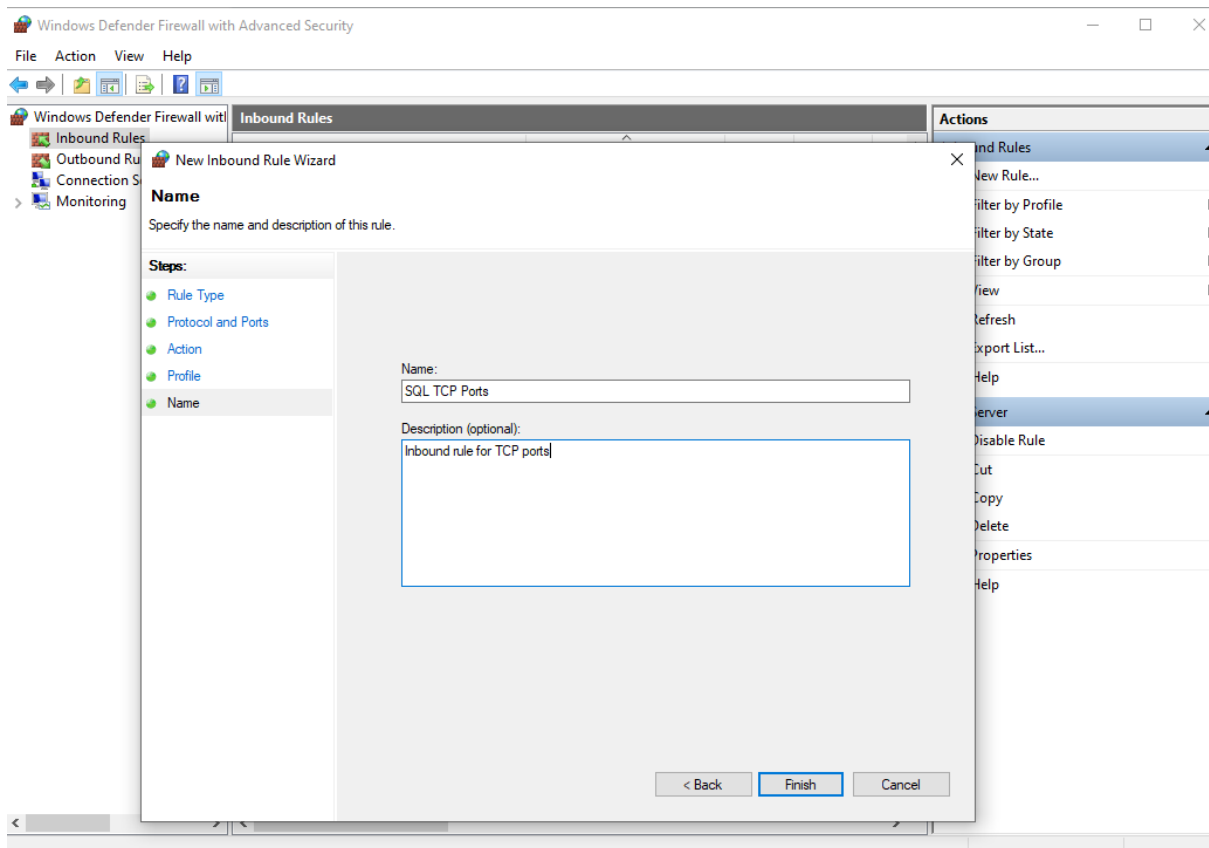


## Inbound rule for TCP ports









## Configuring Windows firewall rules with PowerShell

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\WINDOWS\system32> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Server" -Direction Inbound -Protocol TCP -LocalPort 1433 -Action allow

Name                : {76bde8f3-2071-4dc1-9414-6435a96e8cf5}
DisplayName          : SQL Server
Description          :
DisplayGroup         :
Group               :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Inbound
Action               : Allow
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : The rule was parsed successfully from the store. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource     : PersistentStore
PolicyStoreSourceType : Local

PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Admin Connection" -Direction Inbound -Protocol TCP -LocalPort 1434 -Action allow

Name                : {e6af9125-6437-4c91-9e8b-9ee0543e3c25}
DisplayName          : SQL Admin Connection
Description          :
DisplayGroup         :
Group               :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Inbound
Action               : Allow
```

```
Select Administrator: Windows PowerShell

EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Database Management" -Direction Inbound -Protocol UDP -LocalPort 1434 -Action allow

Name : {94b52db1-258d-4d7b-a7c1-c04cf2d18e83}
DisplayName : SQL Database Management
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Service Broker" -Direction Inbound -Protocol TCP -LocalPort 4022 -Action allow

Name : {9651a802-7903-4820-aea6-90bcef622272}
DisplayName : SQL Service Broker
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
```

```
Select Administrator: Windows PowerShell

LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Debugger/RPC" -Direction Inbound -Protocol TCP -LocalPort 135 -Action allow

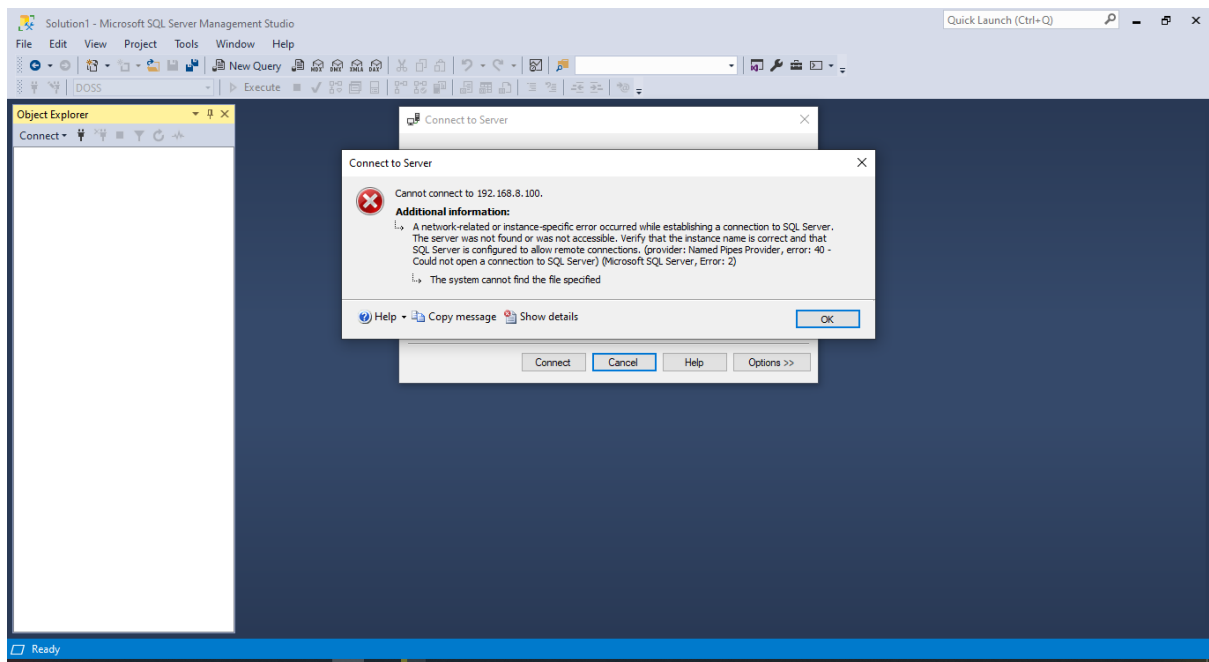
Name : {a4849864-1795-4a47-8847-1cf9a282d321}
DisplayName : SQL Debugger/RPC
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\WINDOWS\system32> #Enabling SQL Analysis Ports
PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "SQL Analysis Services" -Direction Inbound -Protocol TCP -LocalPort 2383 -Action allow

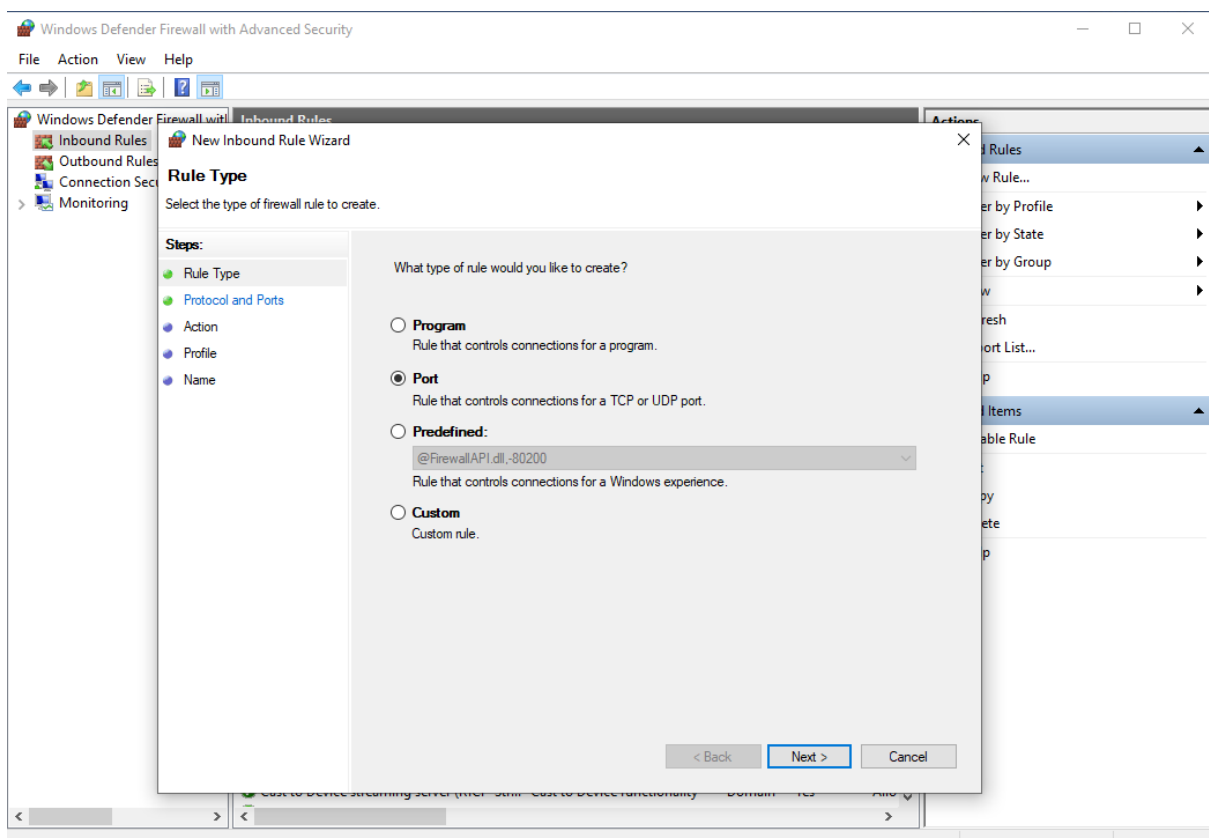
Name : {067d5aae-d5c9-4e23-886f-f4745f1e3a9c}
DisplayName : SQL Analysis Services
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
```

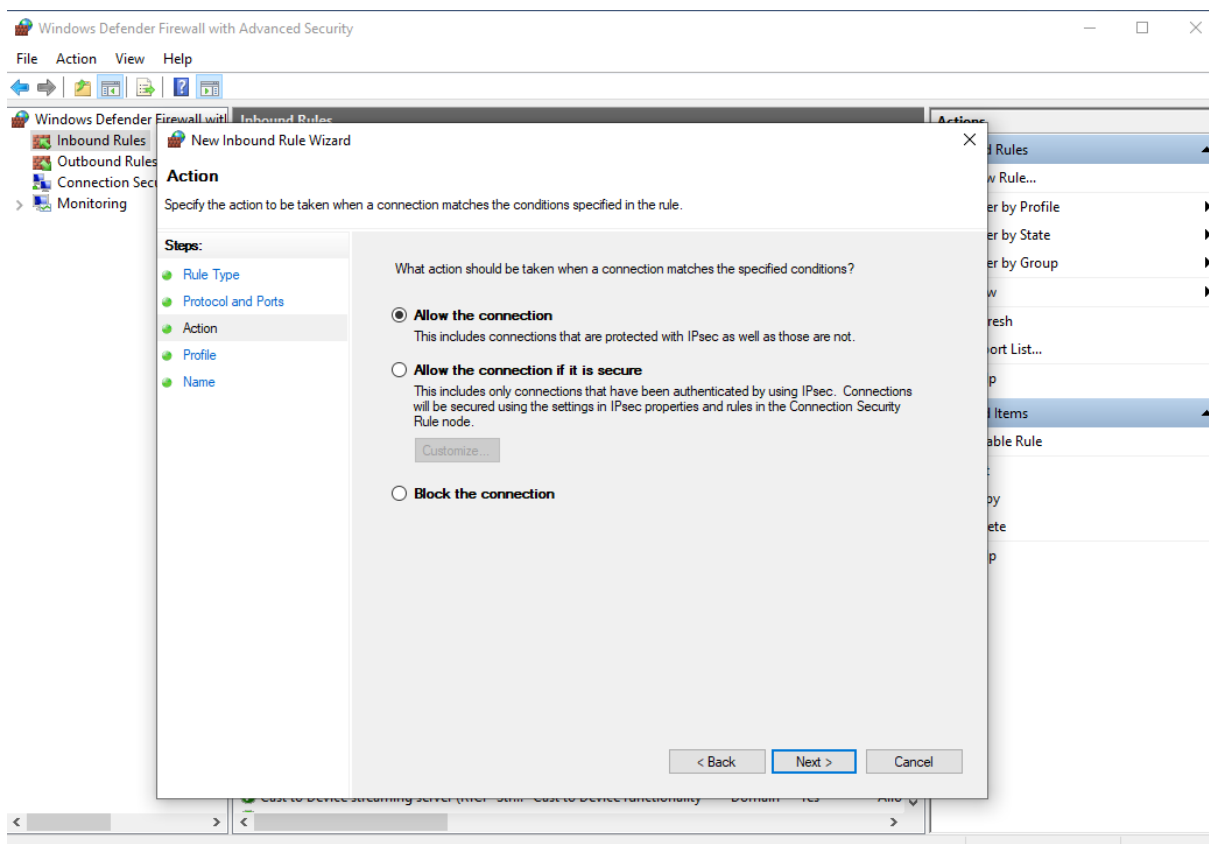
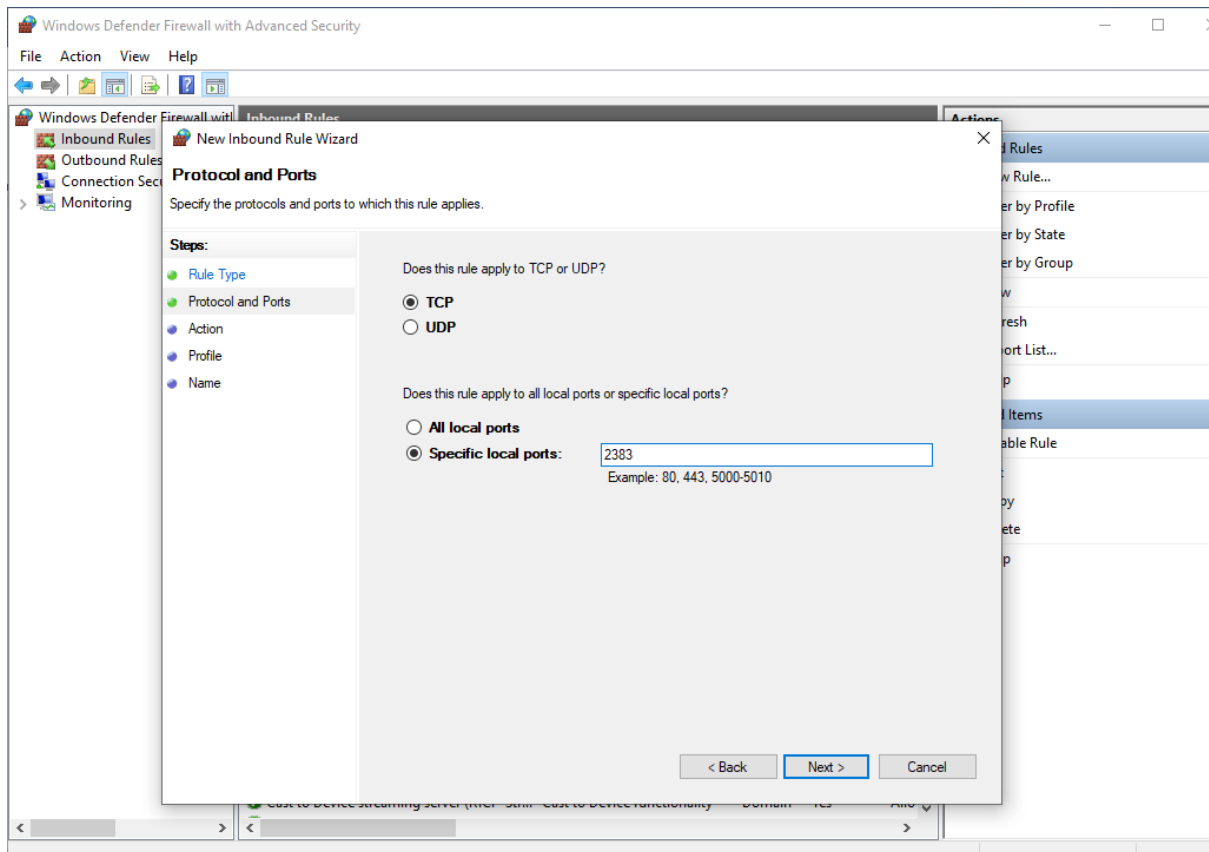


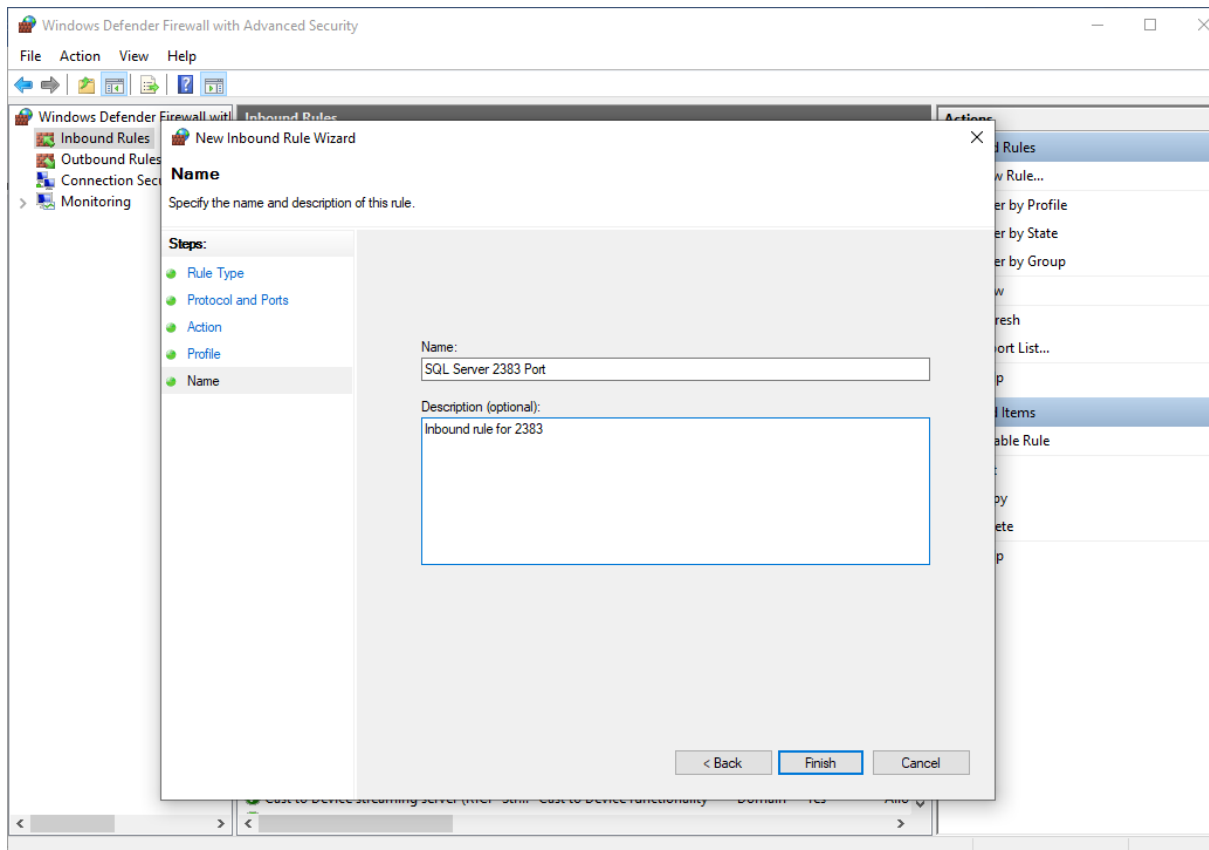
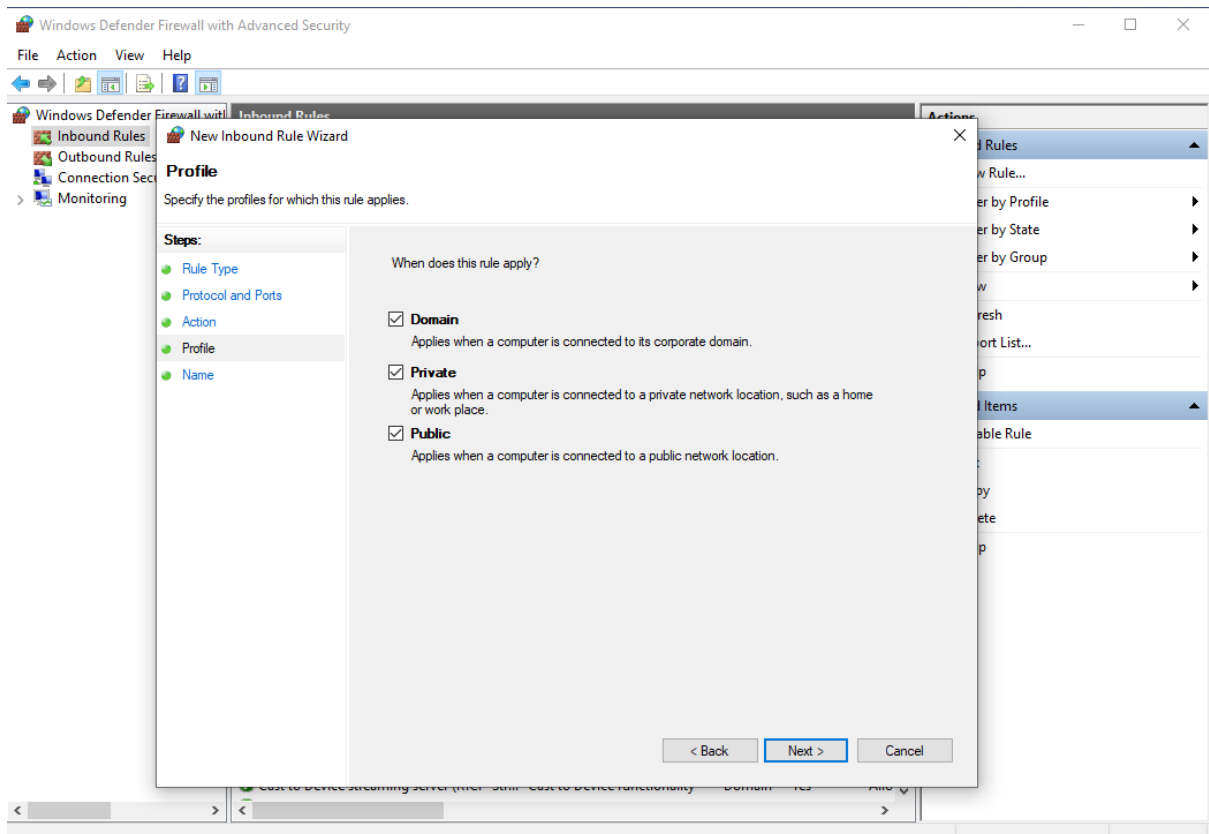
## Trying to log into the DB using the IP / Remote access to the DB



An error has been generated saying not accessible. To solve this an inbound rule should be set for port 2383 a TCP/UDP port.





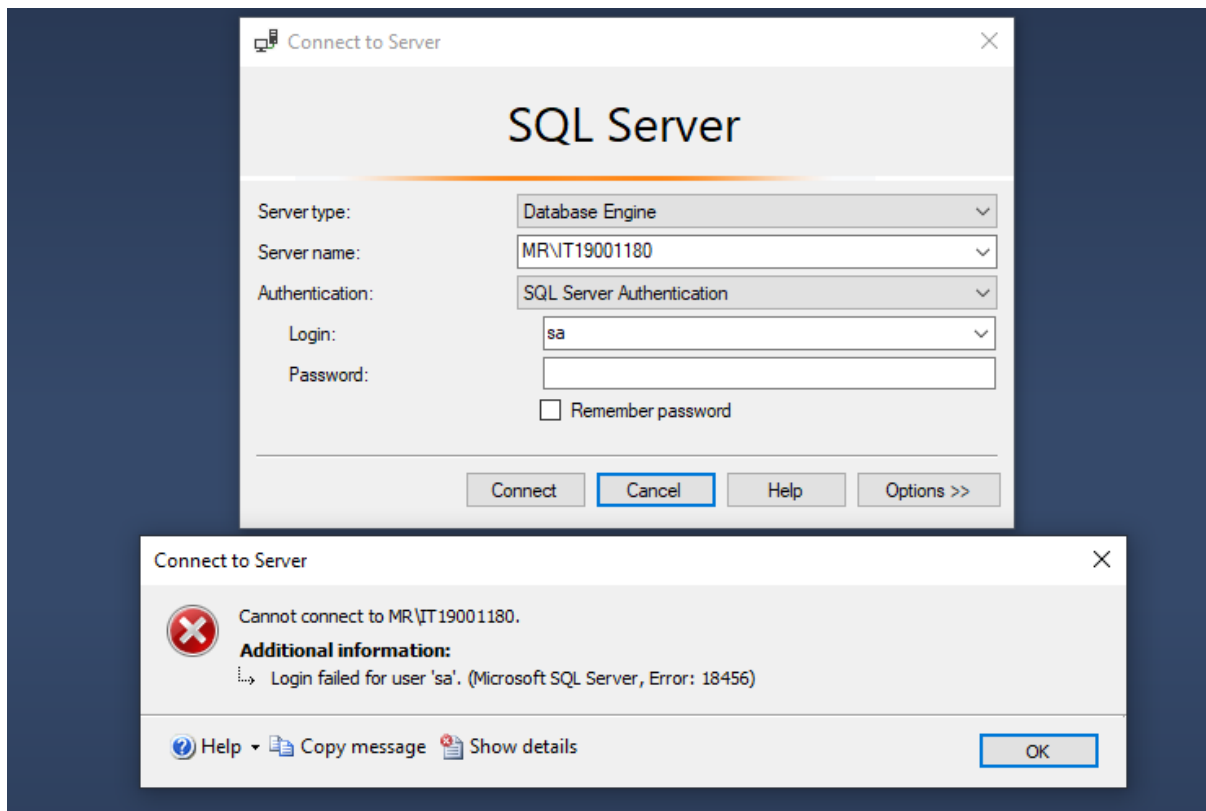


Database Software & Maintain individual login credentials for the people who access the workstation and to perform administrative tasks of the database

- ✓ Default passwords are changed.
- ✓ Null passwords are not used.
- ✓ Individual login credentials.
- ✓ Strong passwords.

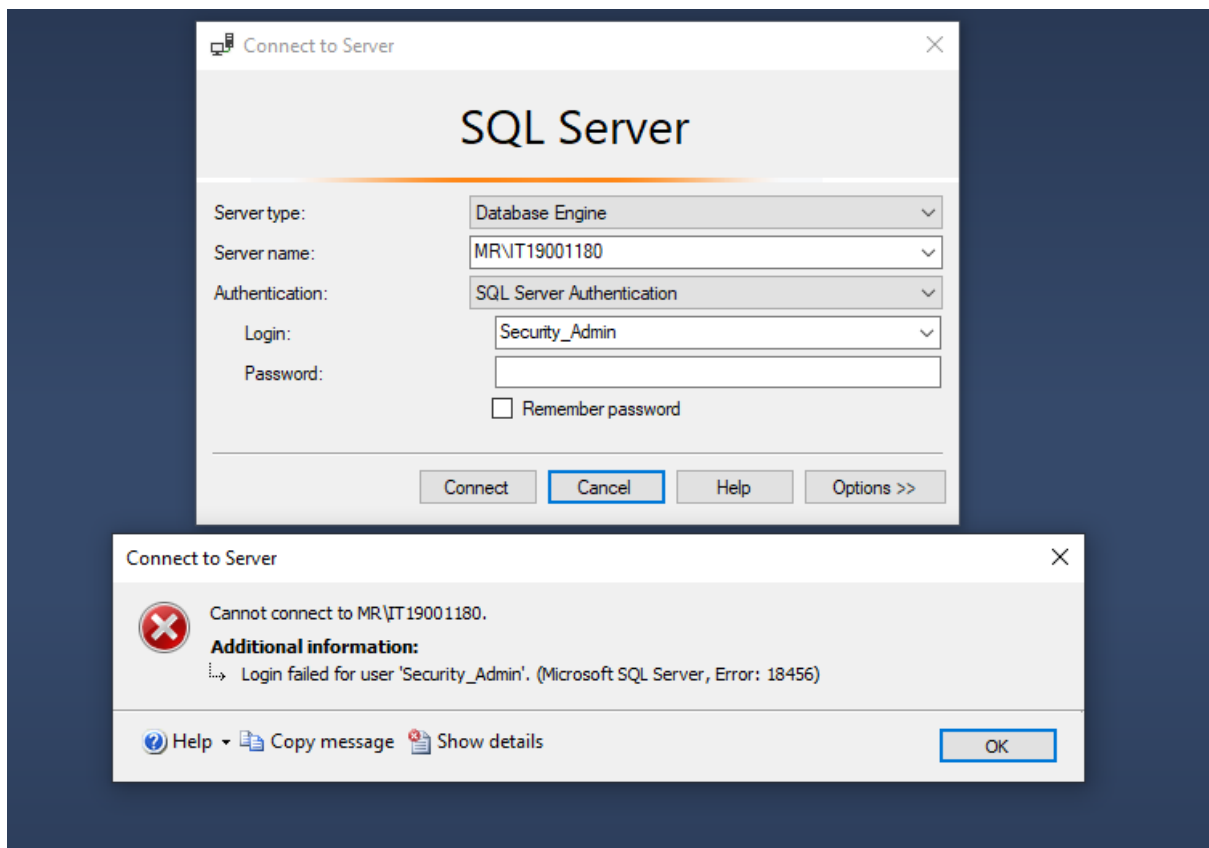
sa – System Administrator

Password: IT19001180



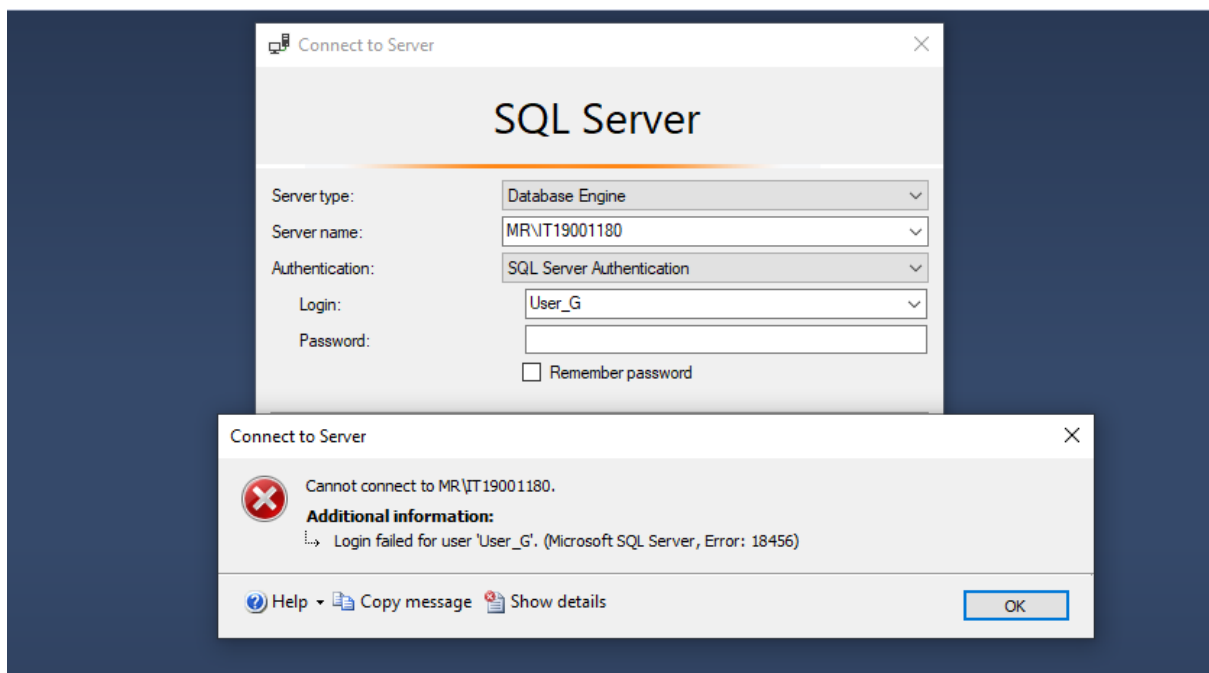
Security\_Admin

Password: SecAdmin



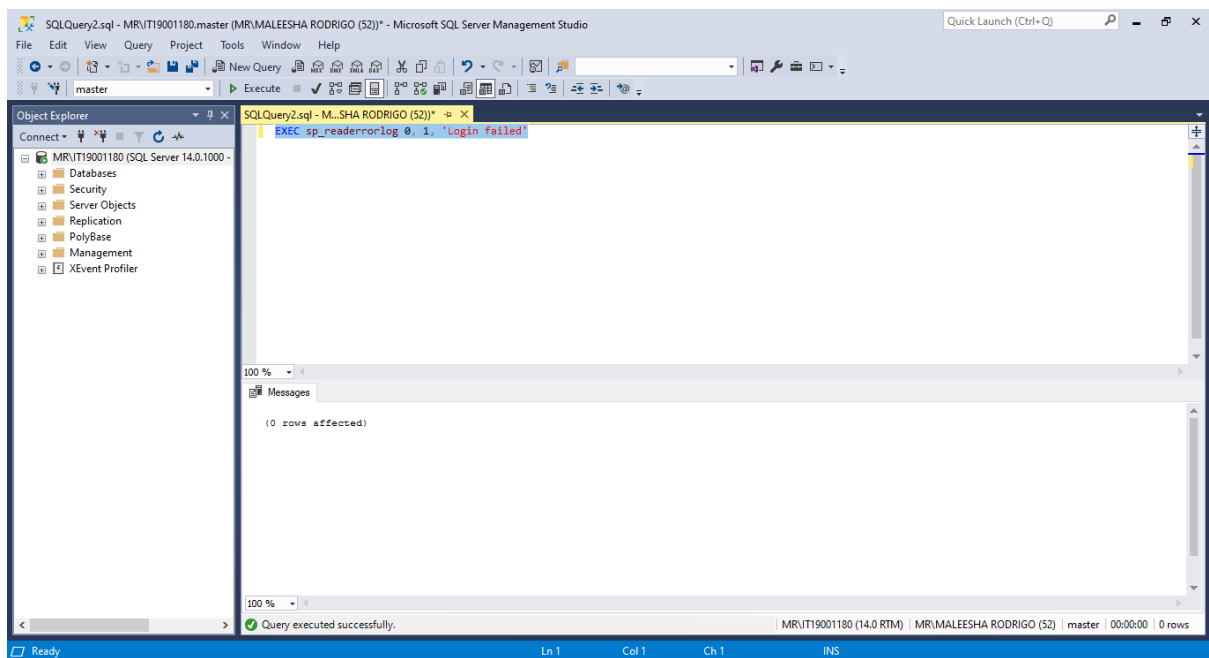
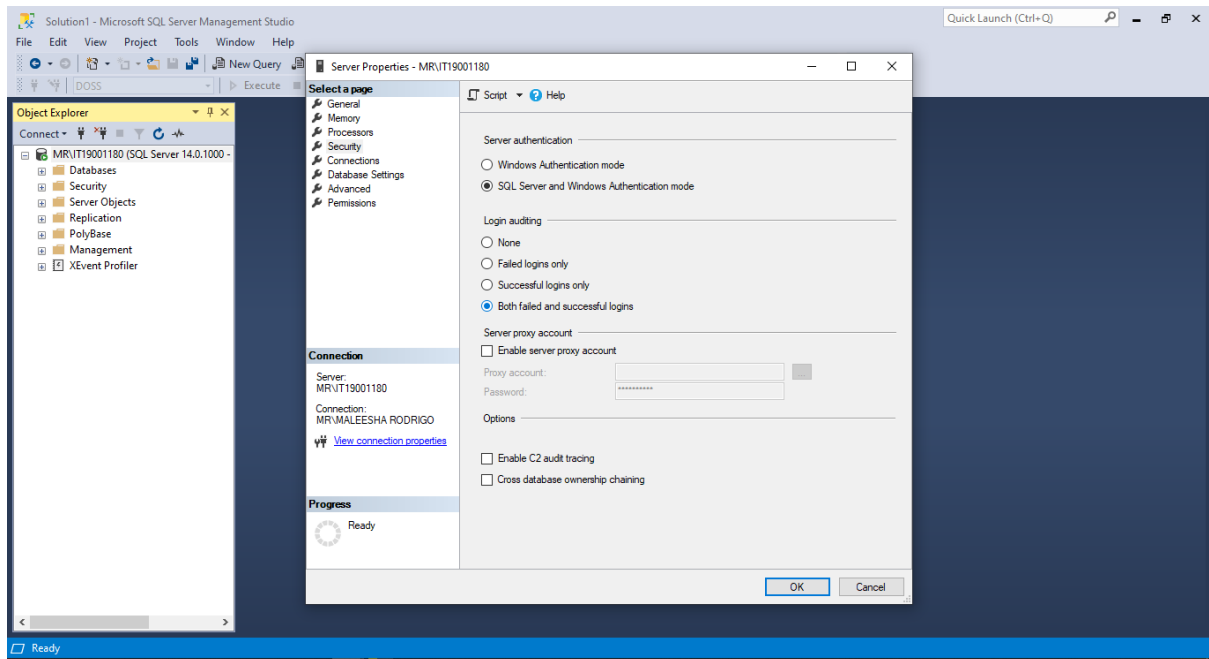
User\_G

Password: Guest2021





Maintain the log records of accessing to the database and maintain the minimum access privileges to the existing servers and applications & Turn on Auditing where technically possible for the database objects with protected data



Logs:

SQLQuery2.sql - MR\UT19001180.master (MR\MALEESHA RODRIGO (52)) - Microsoft SQL Server Management Studio

Object Explorer

- MR\UT19001180 (SQL Server 14.0.1000 -
  - Databases
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

SQLQuery2.sql - M...SHA RODRIGO (52) \*

```
EXEC sp_readerrorlog 0, 1
```

Results

	LogDate	ProcessInfo	Text
1	2021-10-11 18:27:39.800	Server	Microsoft SQL Server 2017 (RTM) - 14.0.1000.169 (X64...
2	2021-10-11 18:27:39.810	Server	UTC adjustment: 5:30
3	2021-10-11 18:27:39.810	Server	(c) Microsoft Corporation.
4	2021-10-11 18:27:39.810	Server	All rights reserved.
5	2021-10-11 18:27:39.810	Server	Server process ID is 1764.
6	2021-10-11 18:27:39.810	Server	System Manufacturer: 'ASUSTeK COMPUTER INC.', Sy...
7	2021-10-11 18:27:39.810	Server	Authentication mode is MIXED.
8	2021-10-11 18:27:39.810	Server	Logging SQL Server messages in file C:\Program Files\...
9	2021-10-11 18:27:39.810	Server	The service account is 'NT Service\MSSQL\$IT190011...
10	2021-10-11 18:27:39.810	Server	Registry startup parameters: -d C:\Program Files\Micr...
11	2021-10-11 18:27:39.810	Server	Command Line Startup Parameters: -s "IT19001180"

Query executed successfully.

MR\UT19001180 (14.0 RTM) | MR\MALEESHA RODRIGO (52) | master | 00:00:00 | 59 rows

## Server level configuration settings

SQLQuery2.sql - MR\UT19001180.master (MR\MALEESHA RODRIGO (52)) - Microsoft SQL Server Management Studio

Object Explorer

- MR\UT19001180 (SQL Server 14.0.1000 -
  - Databases
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

SQLQuery2.sql - M...SHA RODRIGO (52) \*

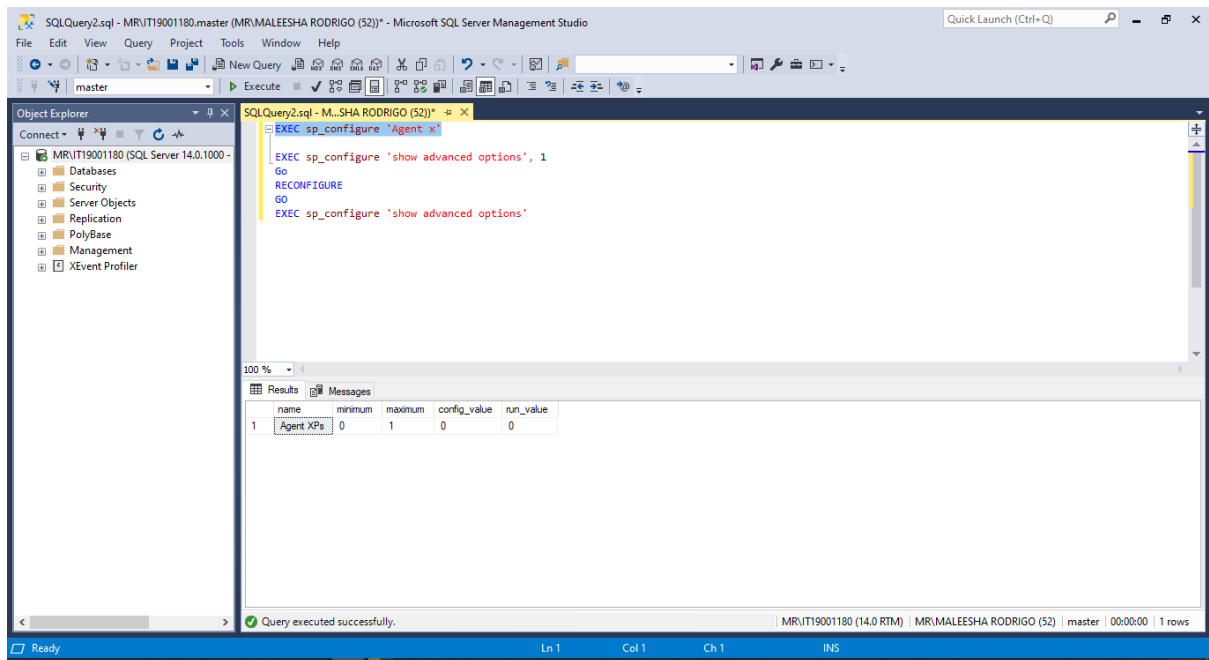
```
EXEC sp_configure 'Agent x'  
  
EXEC sp_configure 'show advanced options', 1  
GO  
RECONFIGURE  
GO  
EXEC sp_configure 'show advanced options'
```

Results

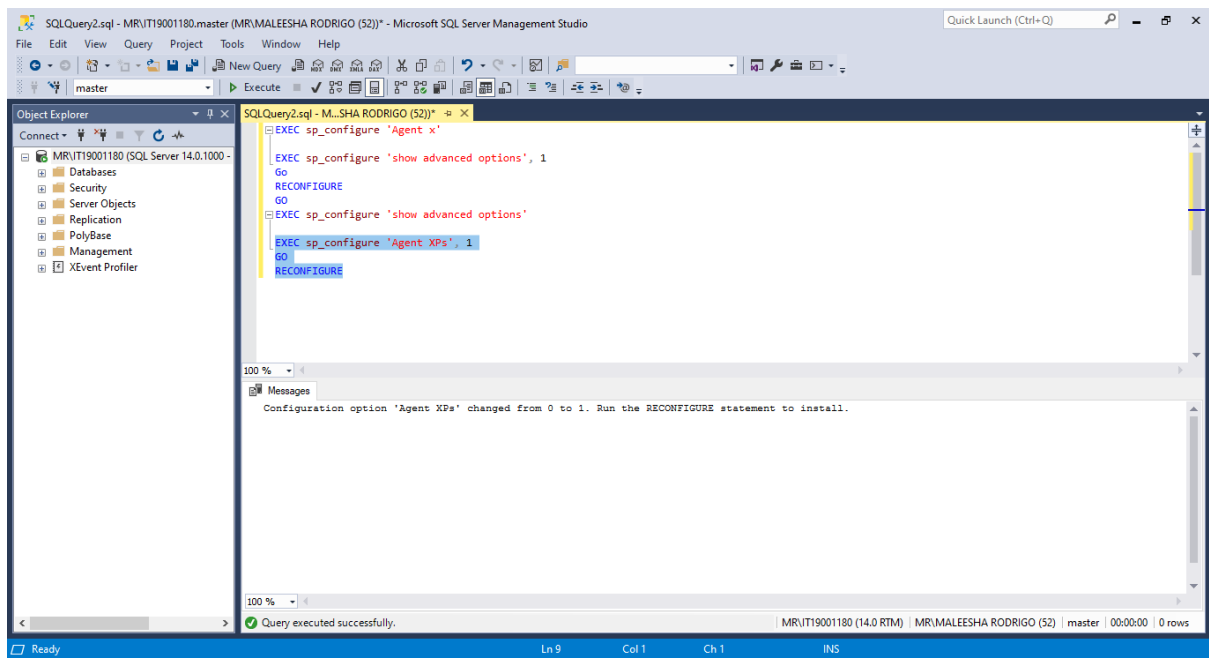
	name	minimum	maximum	config_value	run_value
1	show advanced options	0	1	1	1

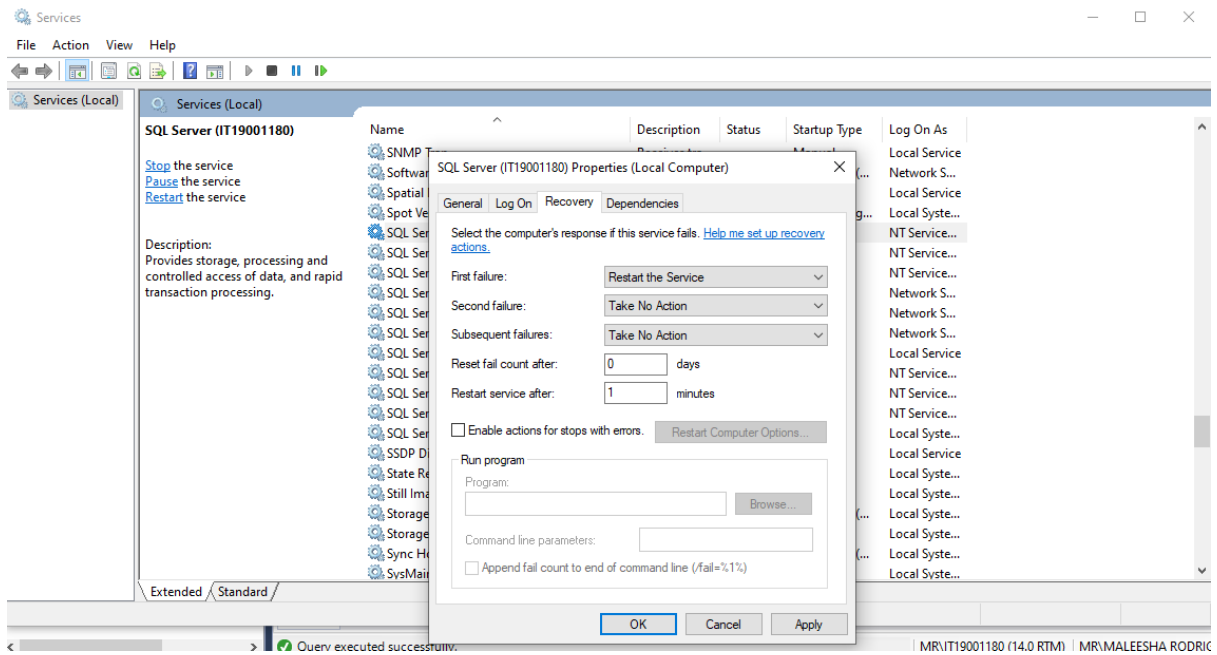
Query executed successfully.

MR\UT19001180 (14.0 RTM) | MR\MALEESHA RODRIGO (52) | master | 00:00:00 | 1 rows



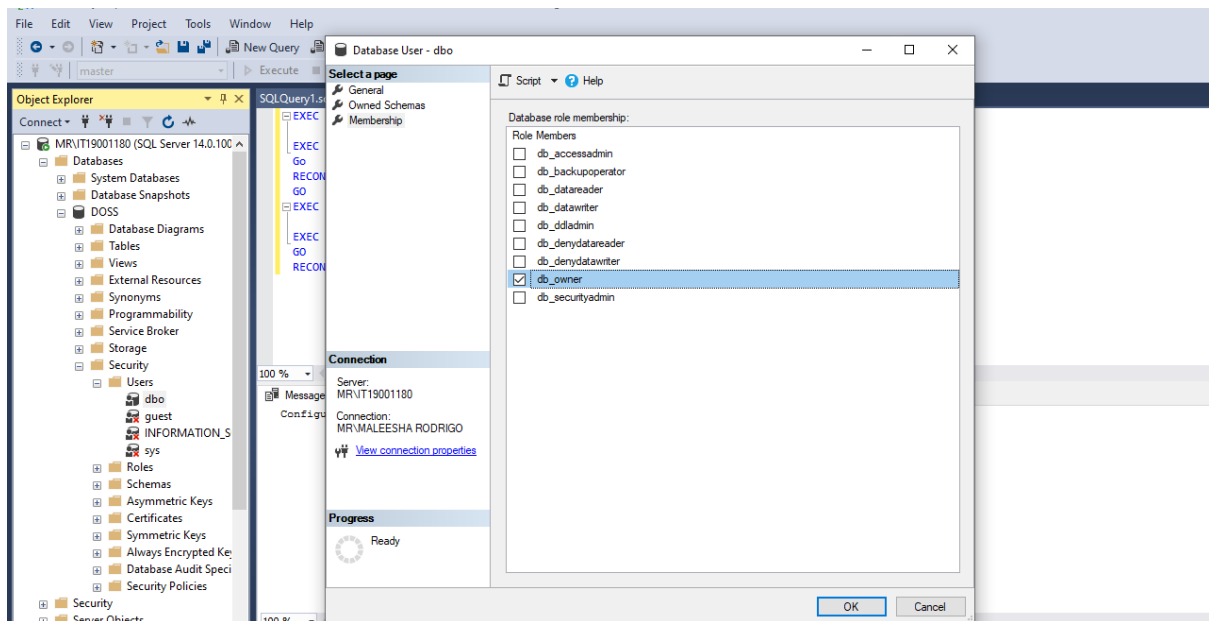
**SQL Server Agent** is needed to execute scheduled tasks on the SQL server. Use the Agent XPs option to enable the SQL Server Agent extended stored procedures on this server. When this option is not enabled, the SQL Server Agent node is not available in SQL Server Management Studio Object Explorer.



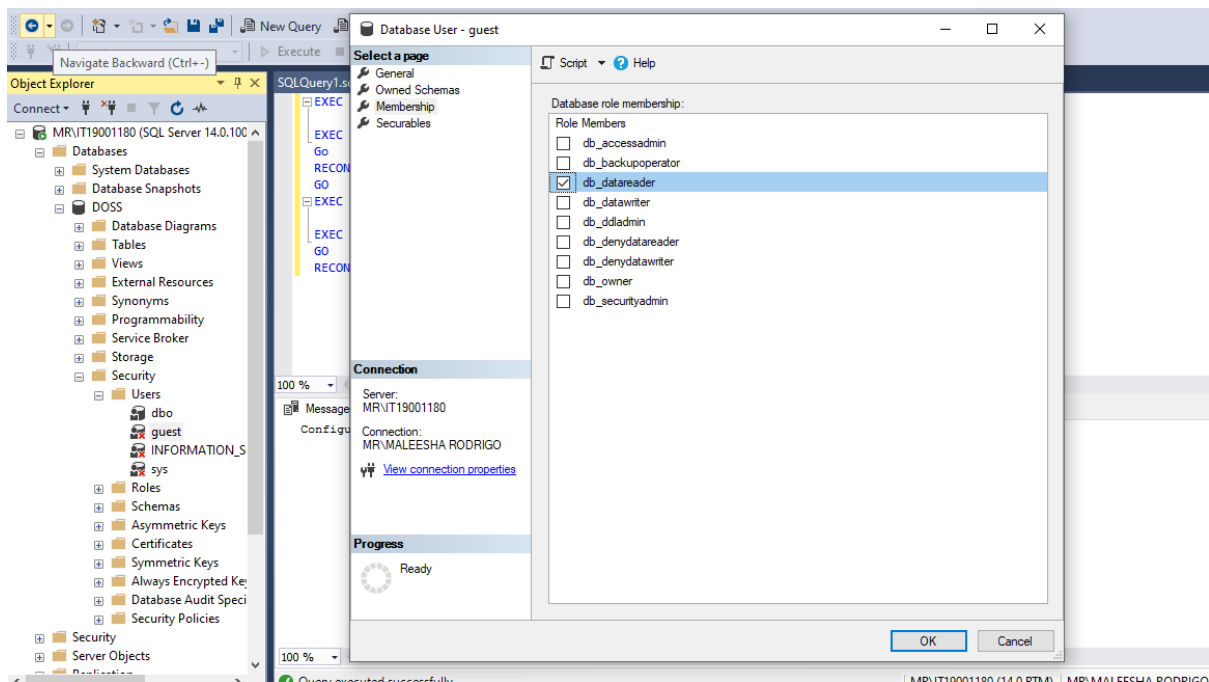


Grant minimal permissions that necessary for the people according to their job role in the database

dbo – DB Owner

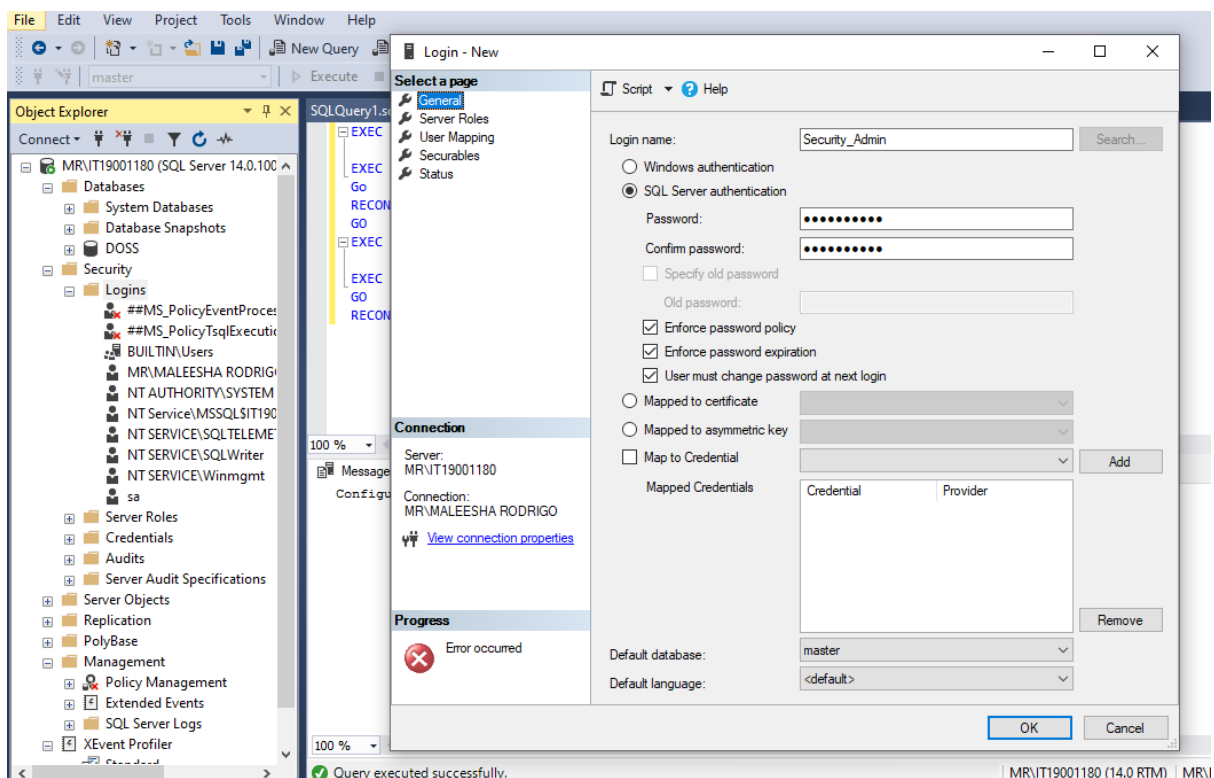


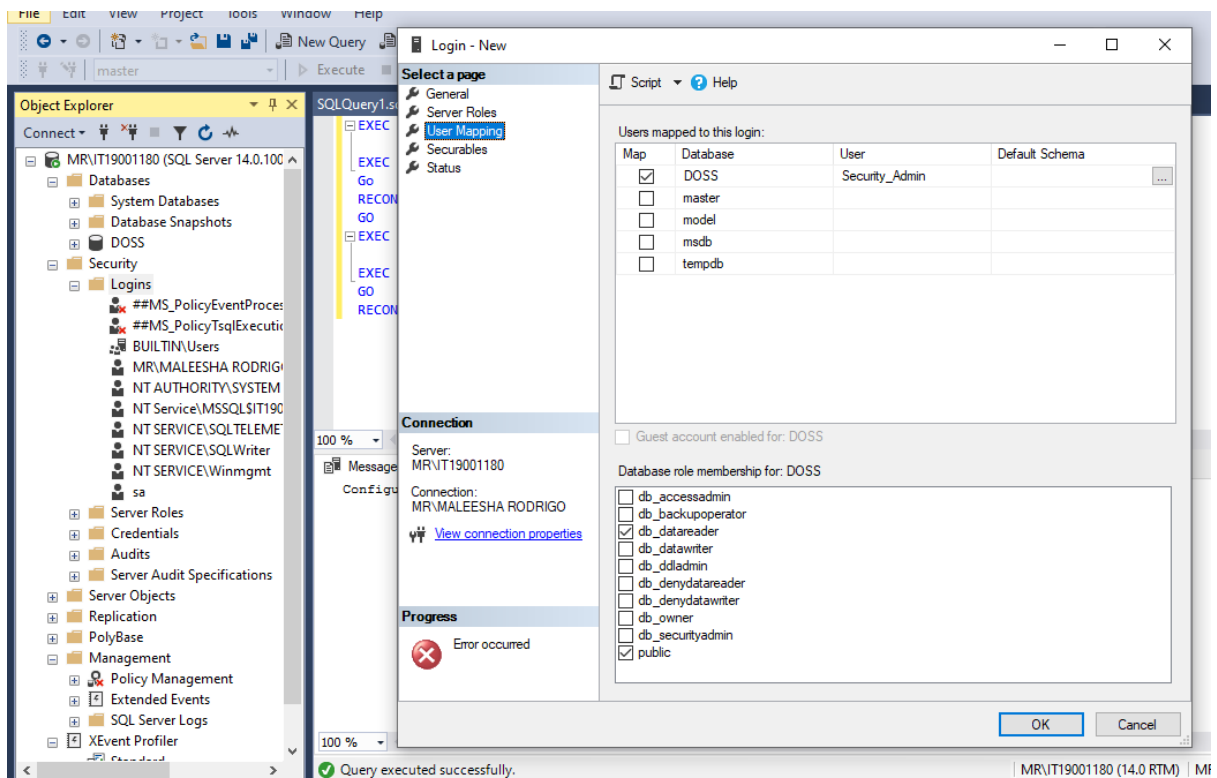
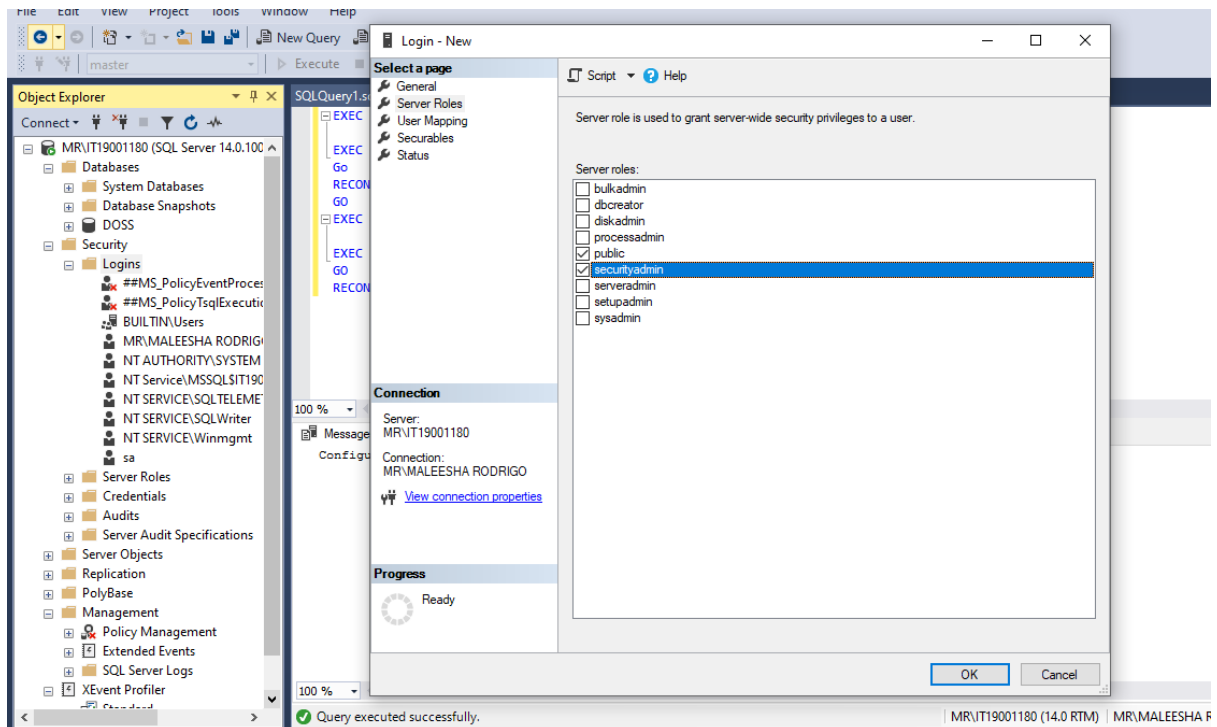
## Guest

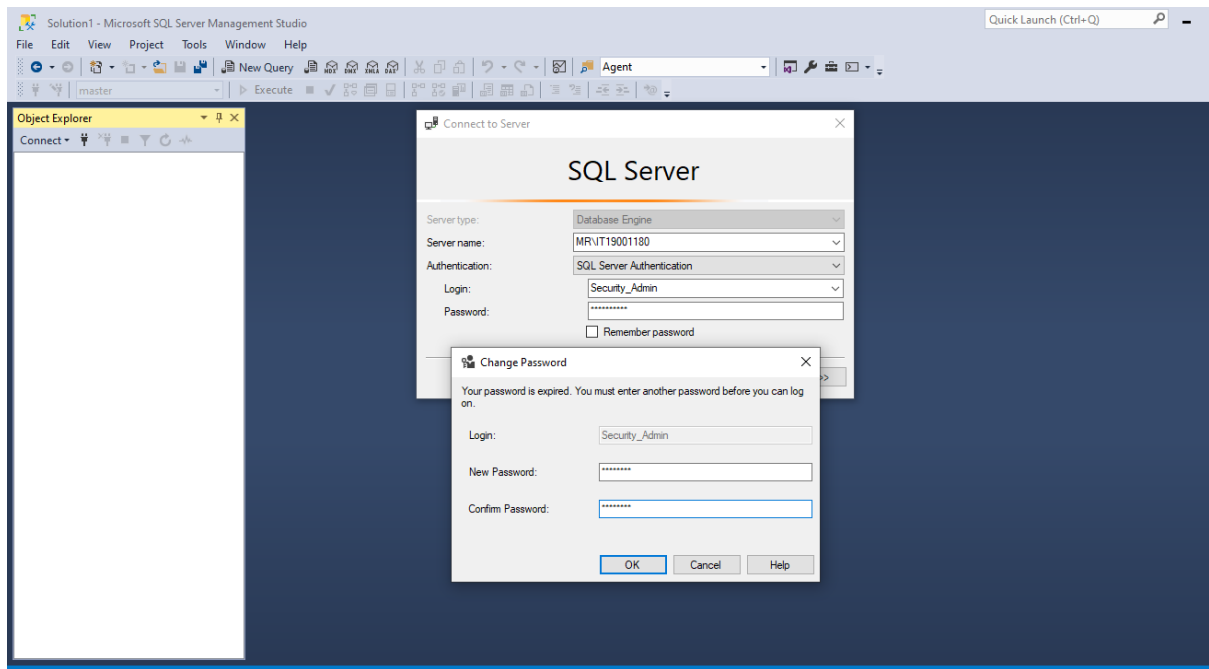


Permissions should be managed through roles or groups and not by direct grants to User IDs where possible

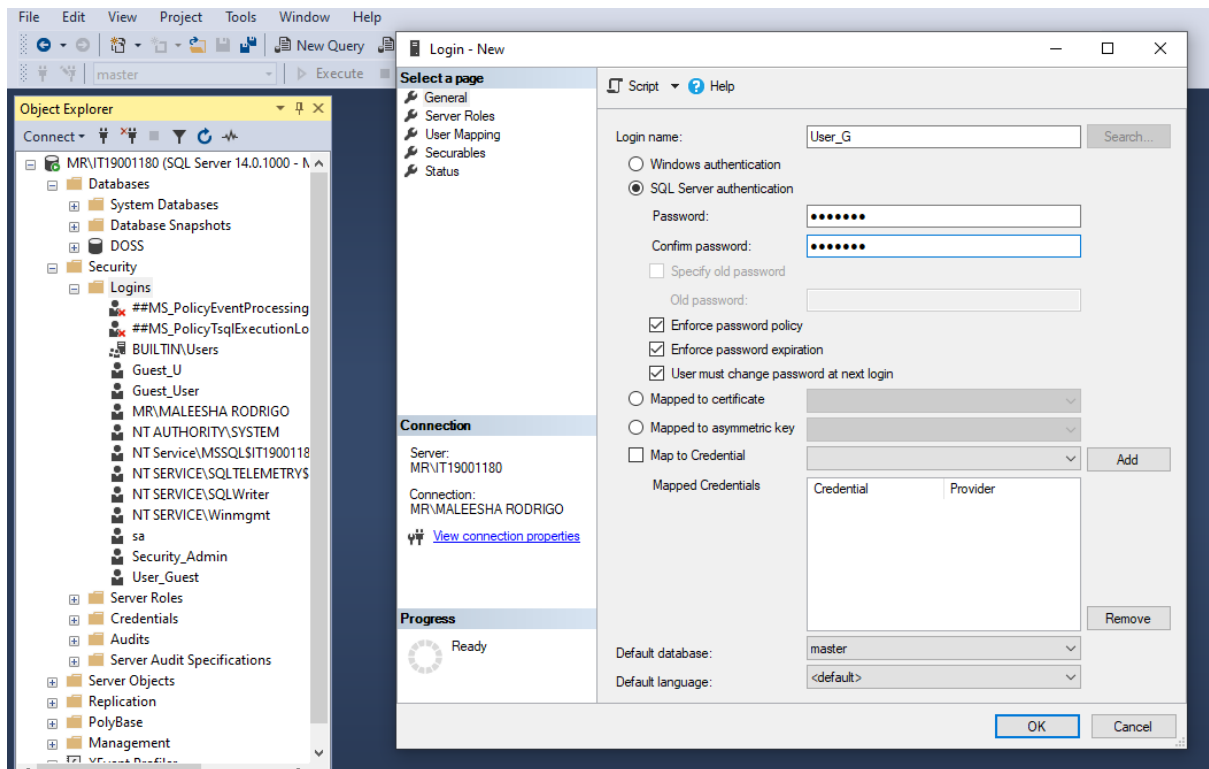
## Security\_Admin

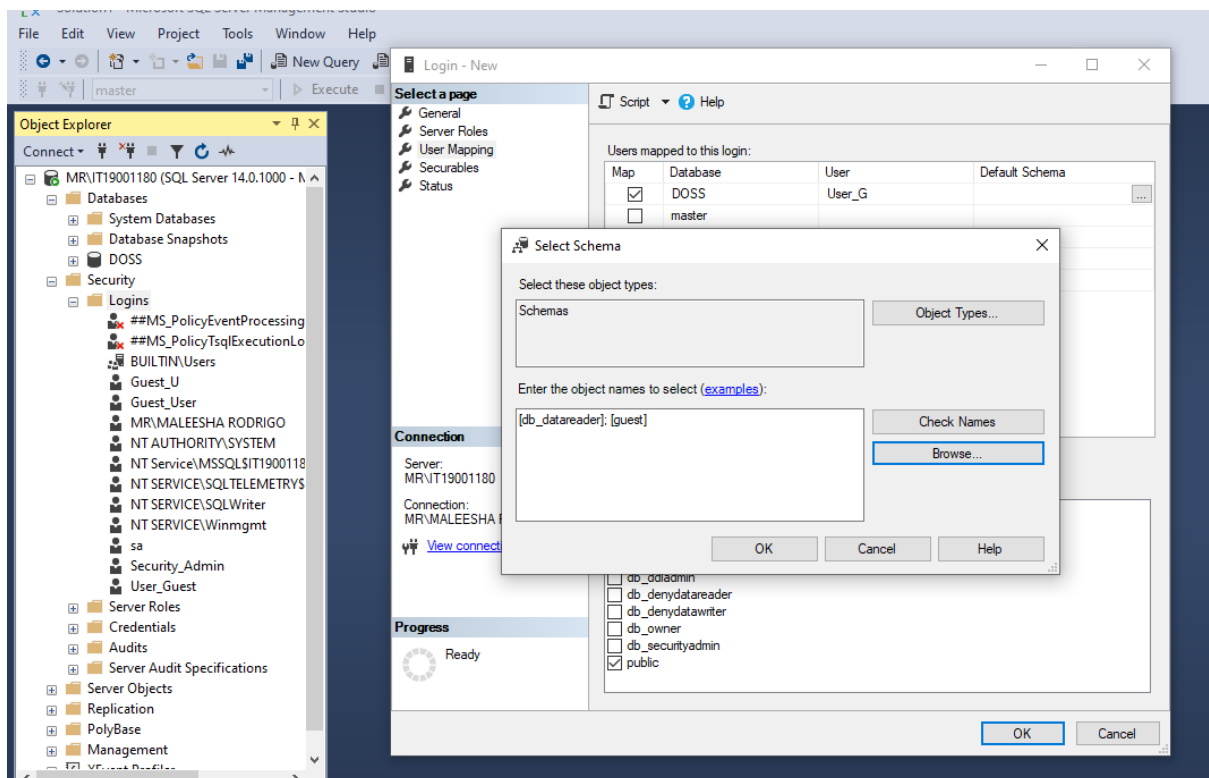
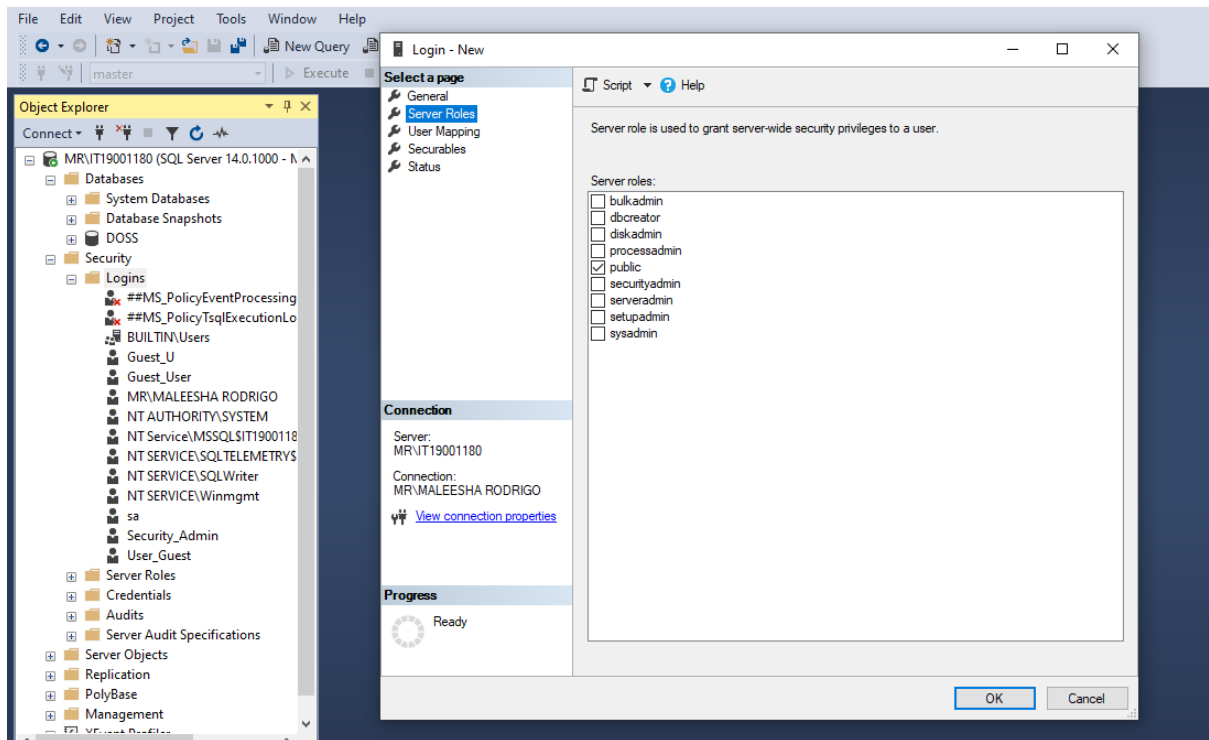




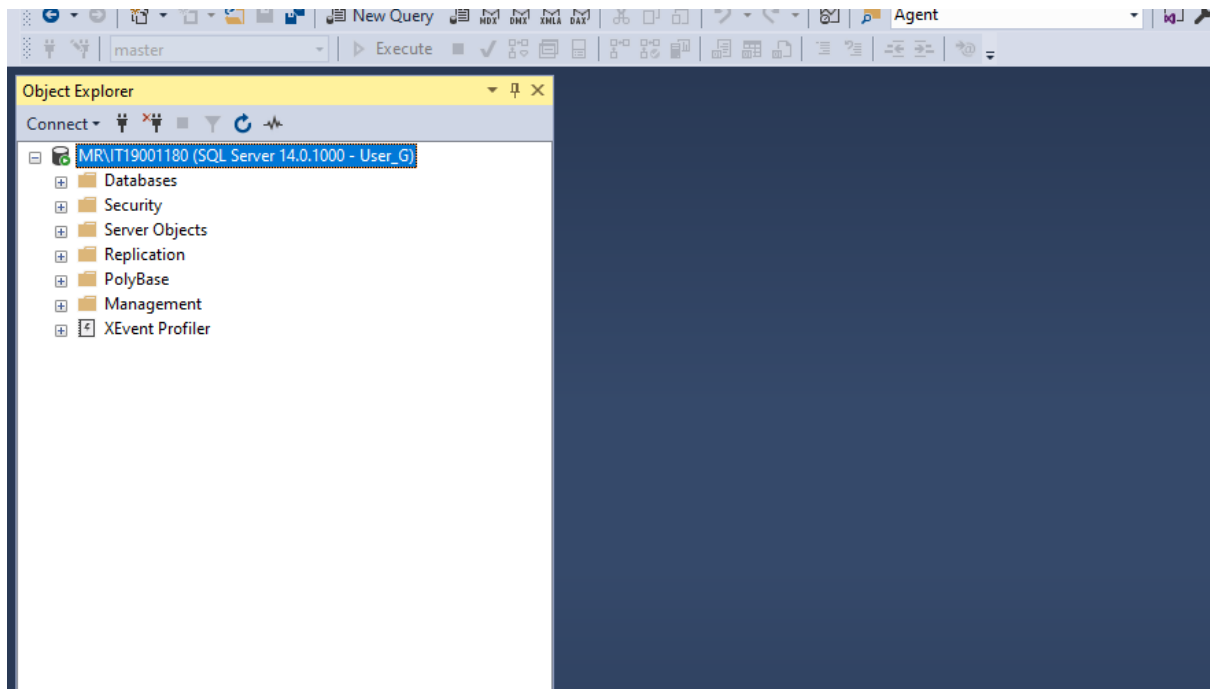


## User\_G

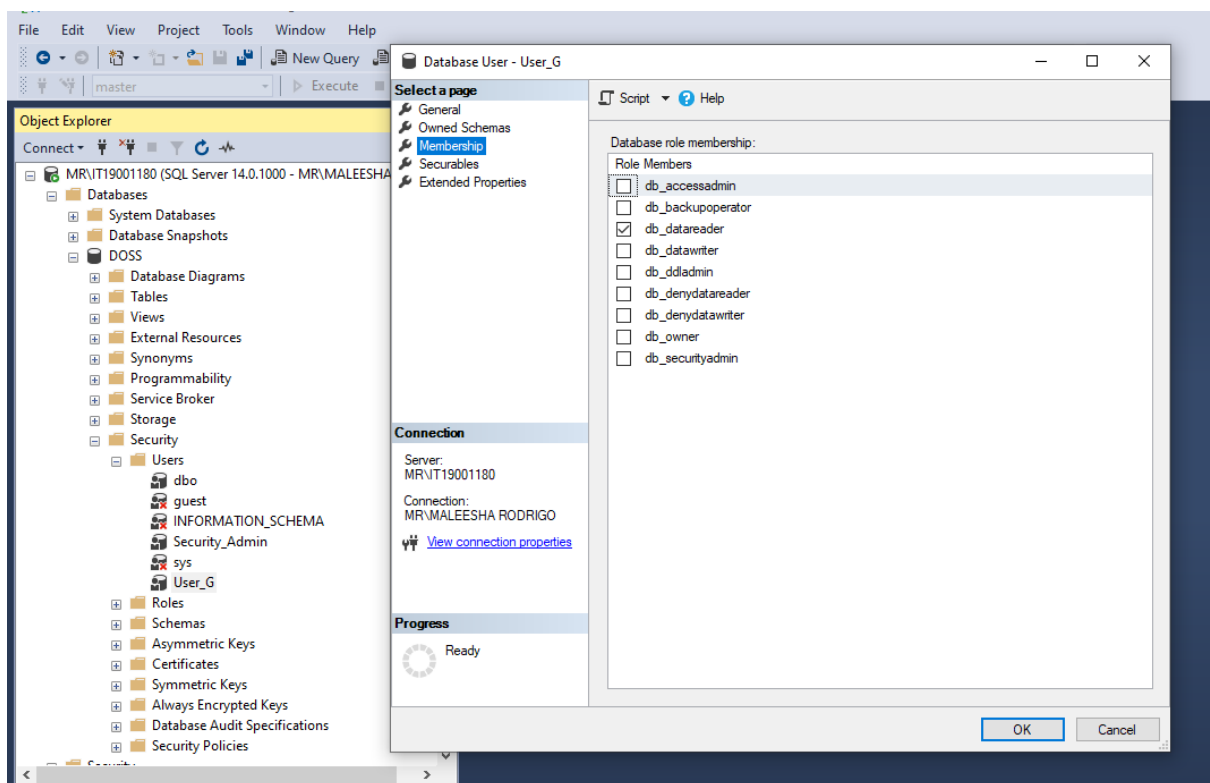






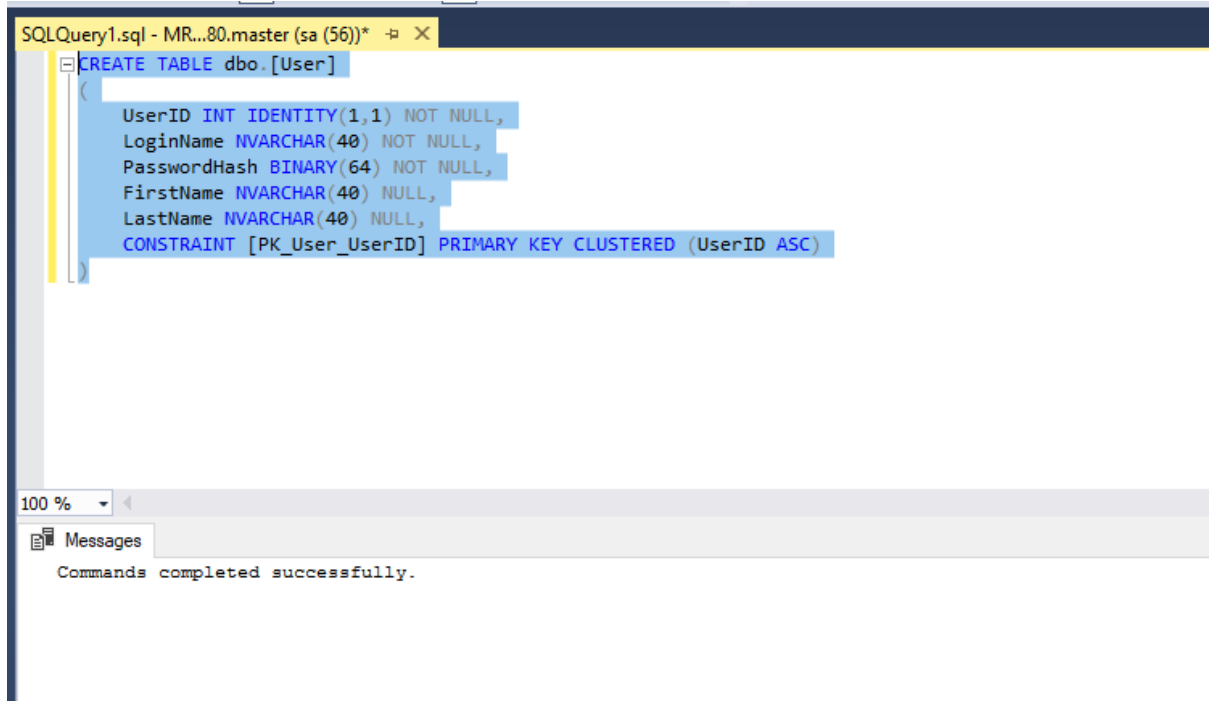


Logged as User\_G.



Manage to use strong password and follow secure methods to preserve the stored passwords

## Hashing



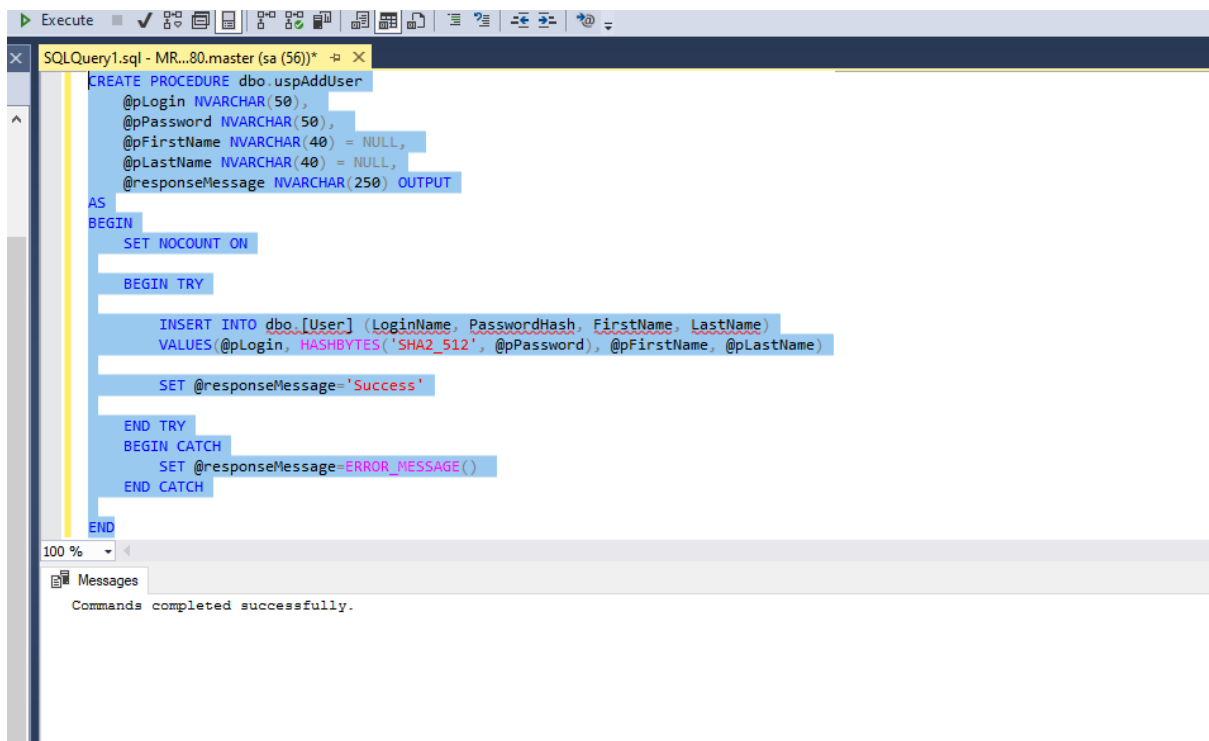
The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query window titled "SQLQuery1.sql - MR...80.master (sa (56))\*". The query is a CREATE TABLE statement for a table named "User" in the "dbo" schema. The table has five columns: "UserID" (INT, IDENTITY(1,1), NOT NULL), "LoginName" (NVARCHAR(40), NOT NULL), "PasswordHash" (BINARY(64), NOT NULL), "FirstName" (NVARCHAR(40), NULL), and "LastName" (NVARCHAR(40), NULL). A primary key constraint named "PK\_User\_UserID" is defined on the "UserID" column. The bottom pane shows the "Messages" tab with the message "Commands completed successfully."

```
CREATE TABLE dbo.[User]
(
    UserID INT IDENTITY(1,1) NOT NULL,
    LoginName NVARCHAR(40) NOT NULL,
    PasswordHash BINARY(64) NOT NULL,
    FirstName NVARCHAR(40) NULL,
    LastName NVARCHAR(40) NULL,
    CONSTRAINT [PK_User_UserID] PRIMARY KEY CLUSTERED (UserID ASC)
)
```

100 %

Messages

Commands completed successfully.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query window titled "SQLQuery1.sql - MR...80.master (sa (56))\*". The query is a CREATE PROCEDURE statement for a procedure named "uspAddUser" in the "dbo" schema. The procedure has five input parameters: "@pLogin" (NVARCHAR(50)), "@pPassword" (NVARCHAR(50)), "@pFirstName" (NVARCHAR(40) = NULL), "@pLastName" (NVARCHAR(40) = NULL), and "@responseMessage" (NVARCHAR(250) OUTPUT). The procedure body starts with "SET NOCOUNT ON", followed by a "BEGIN TRY" block containing an "INSERT INTO" statement for the "User" table. The "INSERT" statement uses "HASHBYTES('SHA2\_512', @pPassword)" for the "PasswordHash" column. After the insert, "@responseMessage" is set to 'Success'. The "BEGIN TRY" block is followed by a "BEGIN CATCH" block where "@responseMessage" is set to "ERROR\_MESSAGE()". The procedure ends with "END CATCH" and "END". The bottom pane shows the "Messages" tab with the message "Commands completed successfully."

```
CREATE PROCEDURE dbo.uspAddUser
    @pLogin NVARCHAR(50),
    @pPassword NVARCHAR(50),
    @pFirstName NVARCHAR(40) = NULL,
    @pLastName NVARCHAR(40) = NULL,
    @responseMessage NVARCHAR(250) OUTPUT
AS
BEGIN
    SET NOCOUNT ON

    BEGIN TRY

        INSERT INTO dbo.[User] (LoginName, PasswordHash, FirstName, LastName)
        VALUES(@pLogin, HASHBYTES('SHA2_512', @pPassword), @pFirstName, @pLastName)

        SET @responseMessage = 'Success'

    END TRY
    BEGIN CATCH
        SET @responseMessage = ERROR_MESSAGE()
    END CATCH
END
```

100 %

Messages

Commands completed successfully.

```
SQLQuery1.sql - MR...80.master (sa (56))* - X
DECLARE @responseMessage NVARCHAR(250)
EXEC dbo.uspAddUser
    @pLogin = N'Security_Admin',
    @pPassword = N'SecAdmin',
    @pFirstName = N'Security',
    @pLastName = N'Administrator',
    @responseMessage=@responseMessage OUTPUT
```

100 %

Messages

Commands completed successfully.

```
SQLQuery1.sql - MR...80.master (sa (56))* - X
DECLARE @responseMessage NVARCHAR(250)
EXEC dbo.uspAddUser
    @pLogin = N'User_G',
    @pPassword = N'Guest2021',
    @pFirstName = N'General',
    @pLastName = N'User',
    @responseMessage=@responseMessage OUTPUT
```

100 %

Messages

Commands completed successfully.

SQLQuery1.sql - MR...80.master (sa (56))\*

```
DECLARE @responseMessage NVARCHAR(250)
EXEC dbo.uspAddUser
    @pLogin = N'sa',
    @pPassword = N'IT19001180',
    @pFirstName = N'System',
    @pLastName = N'Administrator',
    @responseMessage=@responseMessage OUTPUT
```

100 %

Messages

Commands completed successfully.

SQLQuery1.sql - MR...80.master (sa (56))\*

```
SELECT *
FROM [dbo].[User]
```

100 %

Results

	UserID	LoginName	PasswordHash	FirstName	LastName
1	1	Security_Admin	0xE137D2C72090B0EA2ACA92574189FF6C03EDBFB79782A0...	Security	Administrator
2	2	User_G	0x0F72B204F45DCF39ECF1B2475EDA3736E8F84458B7305D6...	General	User
3	3	sa	0x6AE764377E57BDA8483129846420FE7B6DBFA7DD151920...	System	Administrator

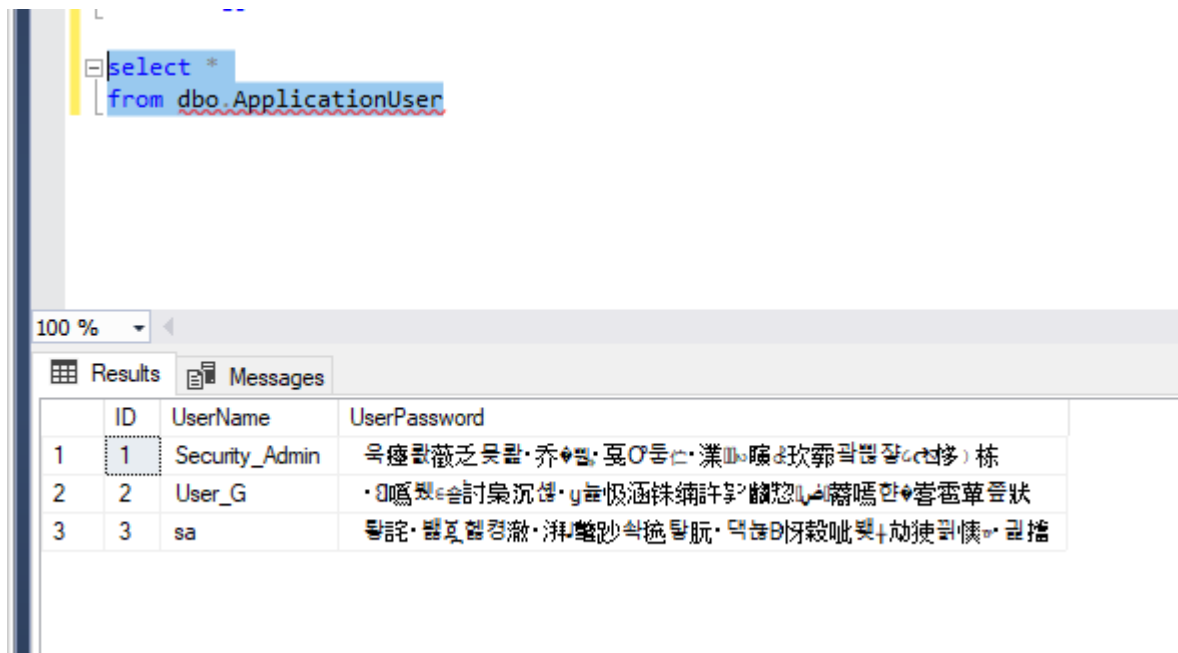
## Encryption

```
SQLQuery1.sql - MR...80.master (sa (56))* - X
CREATE TABLE dbo.ApplicationUser
(
    ID int Identity,
    UserName NVARCHAR(100),
    UserPassword NVARCHAR(100)
)

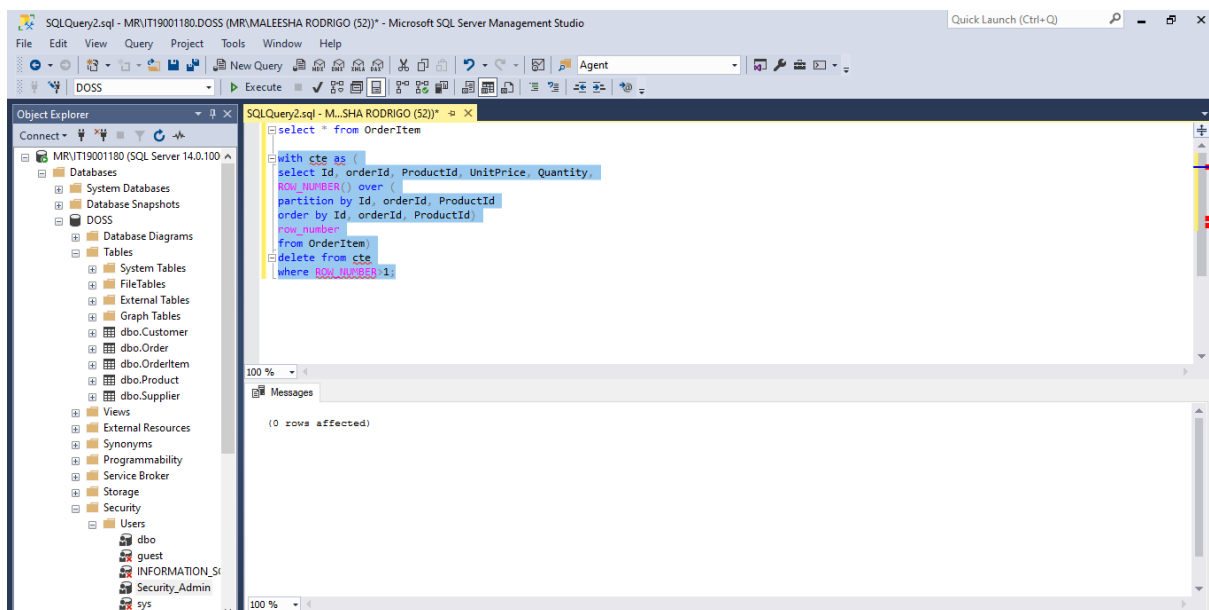
100 %
Messages
Commands completed successfully.
```

```
SQLQuery1.sql - MR...80.master (sa (56))* - X
Insert Into dbo.ApplicationUser
SELECT 'Security_Admin', PWDENCRYPT('SecAdmin')
GO
Insert Into dbo.ApplicationUser
SELECT 'User_G', PWDENCRYPT('Guest2021')
GO
Insert Into dbo.ApplicationUser
SELECT 'sa', PWDENCRYPT('IT19001180')
GO

100 %
Messages
(1 row affected)
(1 row affected)
(1 row affected)
```

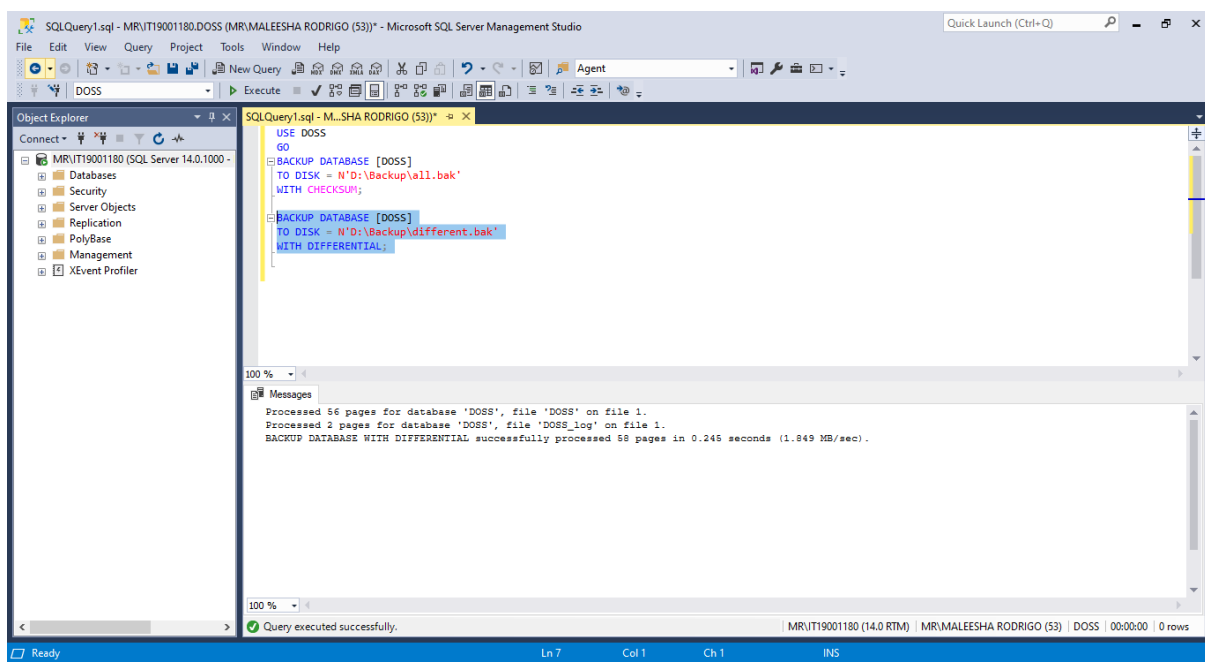
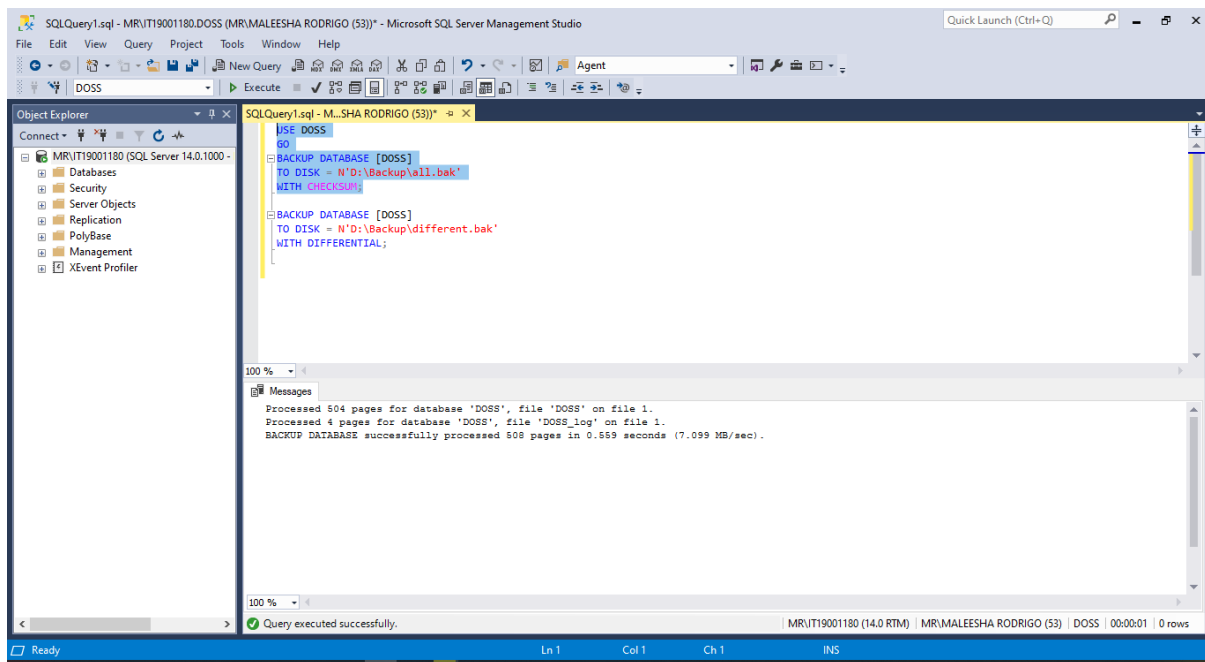


Prevent from redundancy of the stored records of the database



Note: this is a simple DB. Therefore, there are not any redundancy in any table.

## Discuss how manage the implemented database backup and recovery



The screenshot shows a Windows File Explorer window with the address bar set to 'This PC > DATA (D:) > Backup'. The file list contains two files:

Name	Date modified	Type	Size
all.bak	13/10/2021 08:04	BAK File	4,148 KB
different.bak	13/10/2021 08:07	BAK File	564 KB

There are 2 types of backups. A **full backup** is a total copy of the entire data, which backs up all of your files into a single version. A **differential backup** is a growing backup of all files changed since the last backup.

