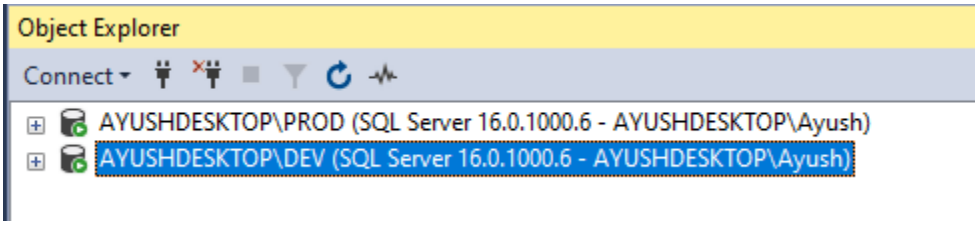
# Mastering Database Refreshes in SQL Server: A Practical Guide

Performing a database refresh from production to development environments is a critical task for maintaining accurate, up-to-date testing environments in SQL Server management. Here, I share my step-by-step approach, including the challenges I faced and the solutions I applied, to ensure a smooth refresh process.

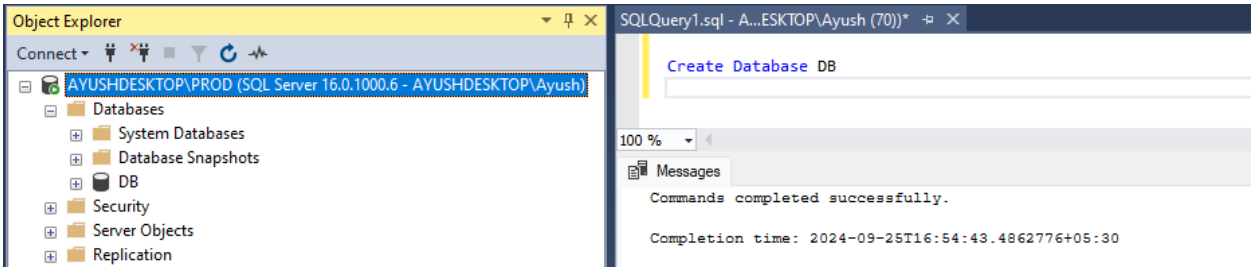
## Step 1 & 2: Established PROD and DEV instances and created folders for backup storage.

- ❑ Created two Instances PROD and DEV

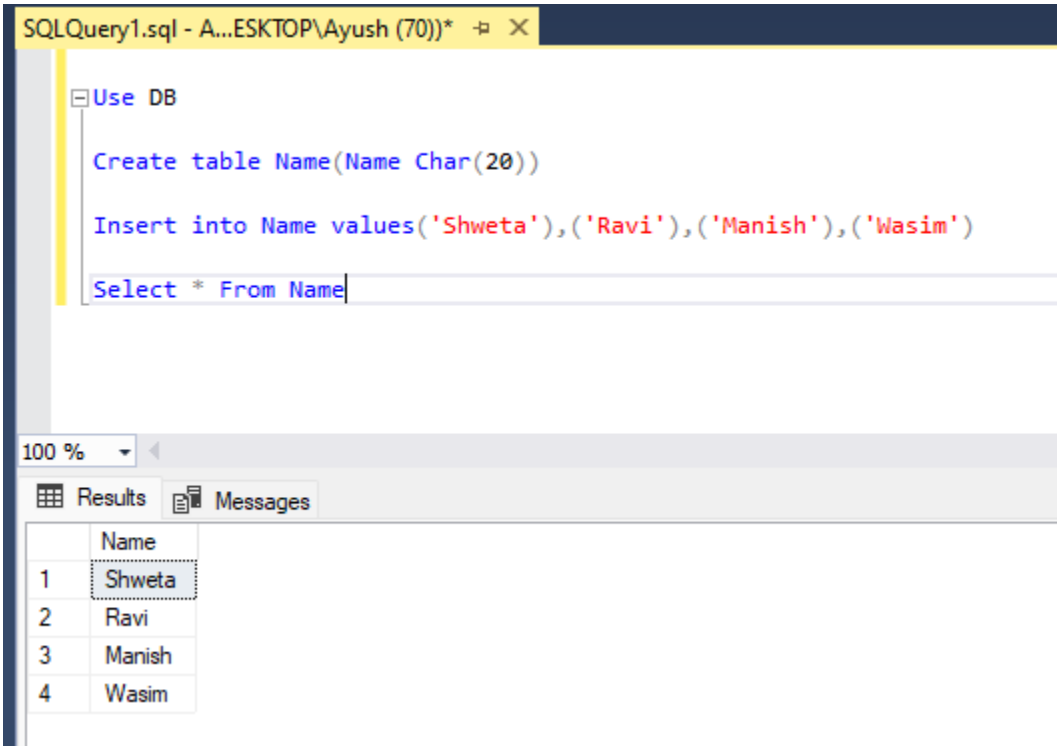



- ❑ Created two folders for Saving the Source Backup and the Destination Backup


## Step 3: Configured a new database on the PROD instance.



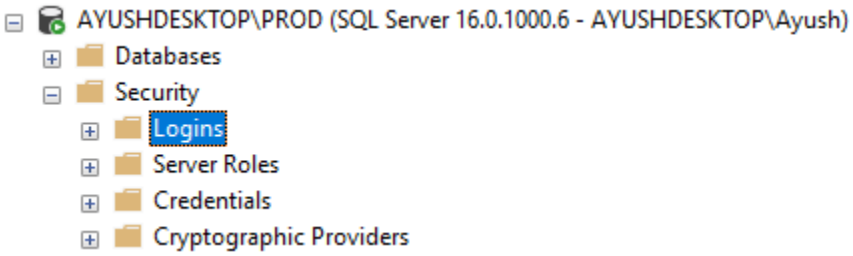

## Step 4: Populated the database with tables and preliminary data.

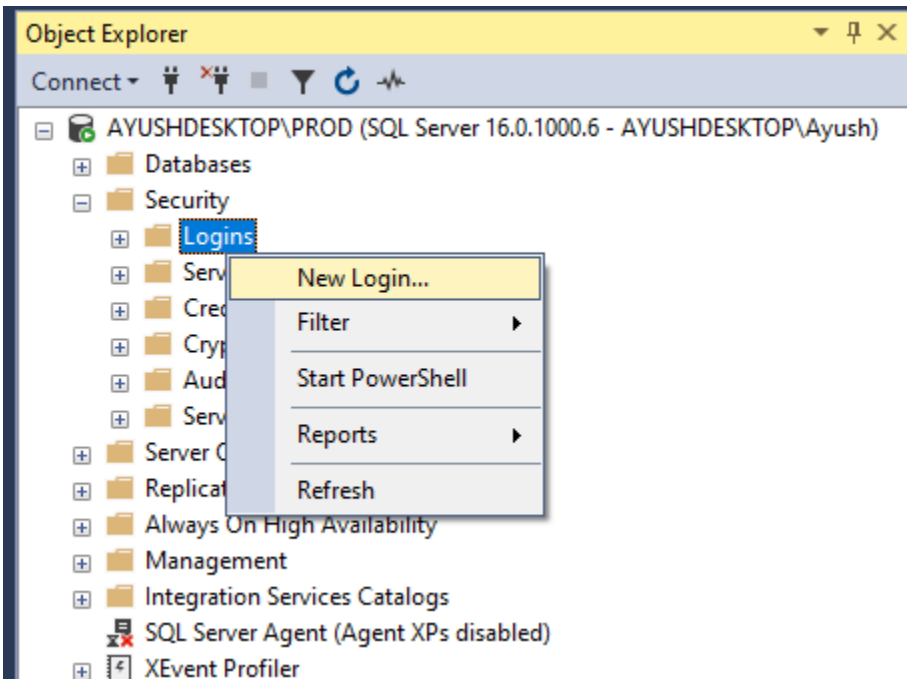



## Steps 5 & 6: Set up instance-level logins and database-level users, assigning specific permissions to control database access effectively.

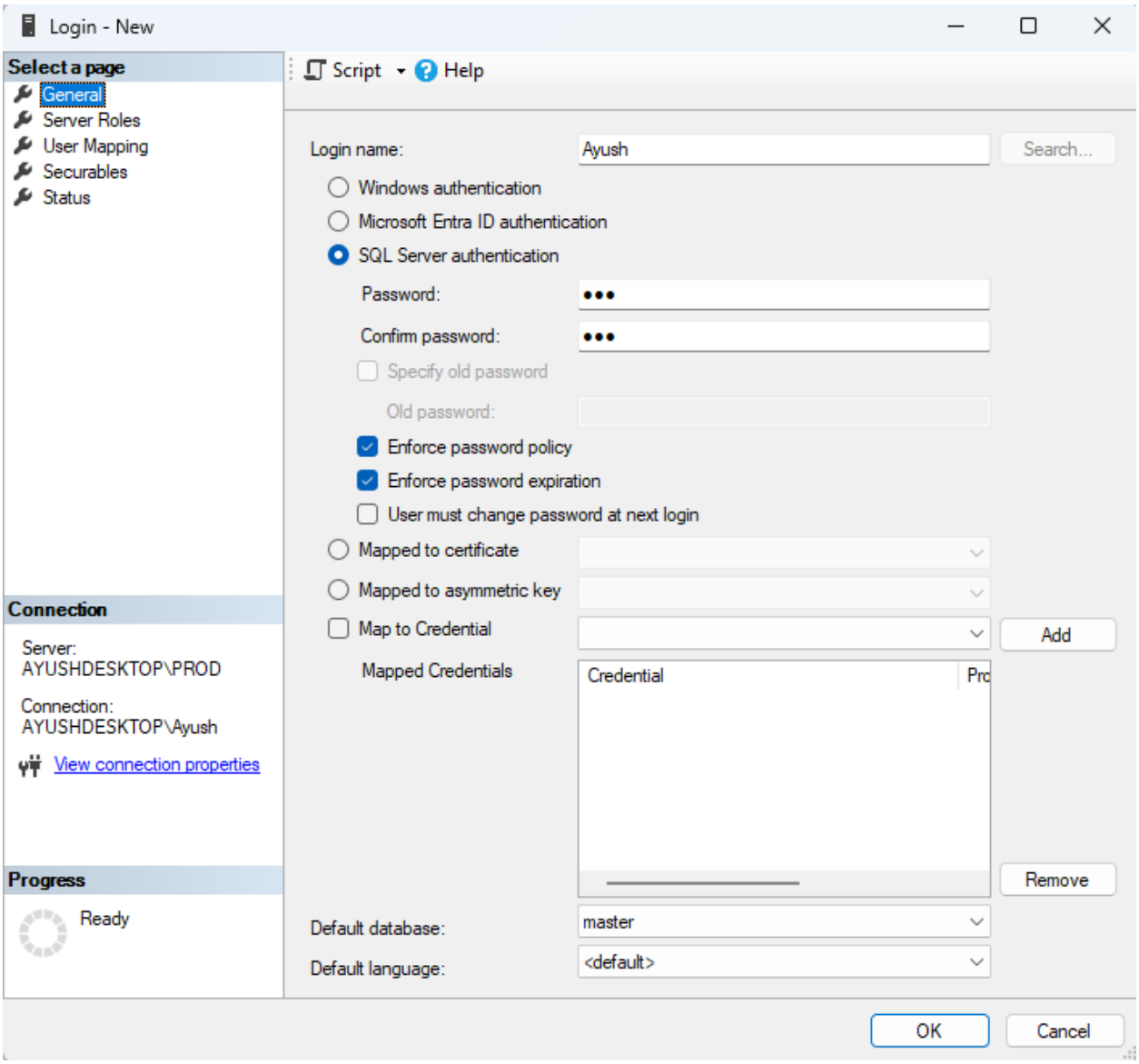
Select the PROD Instance – Expand Security – Right Click on Logins

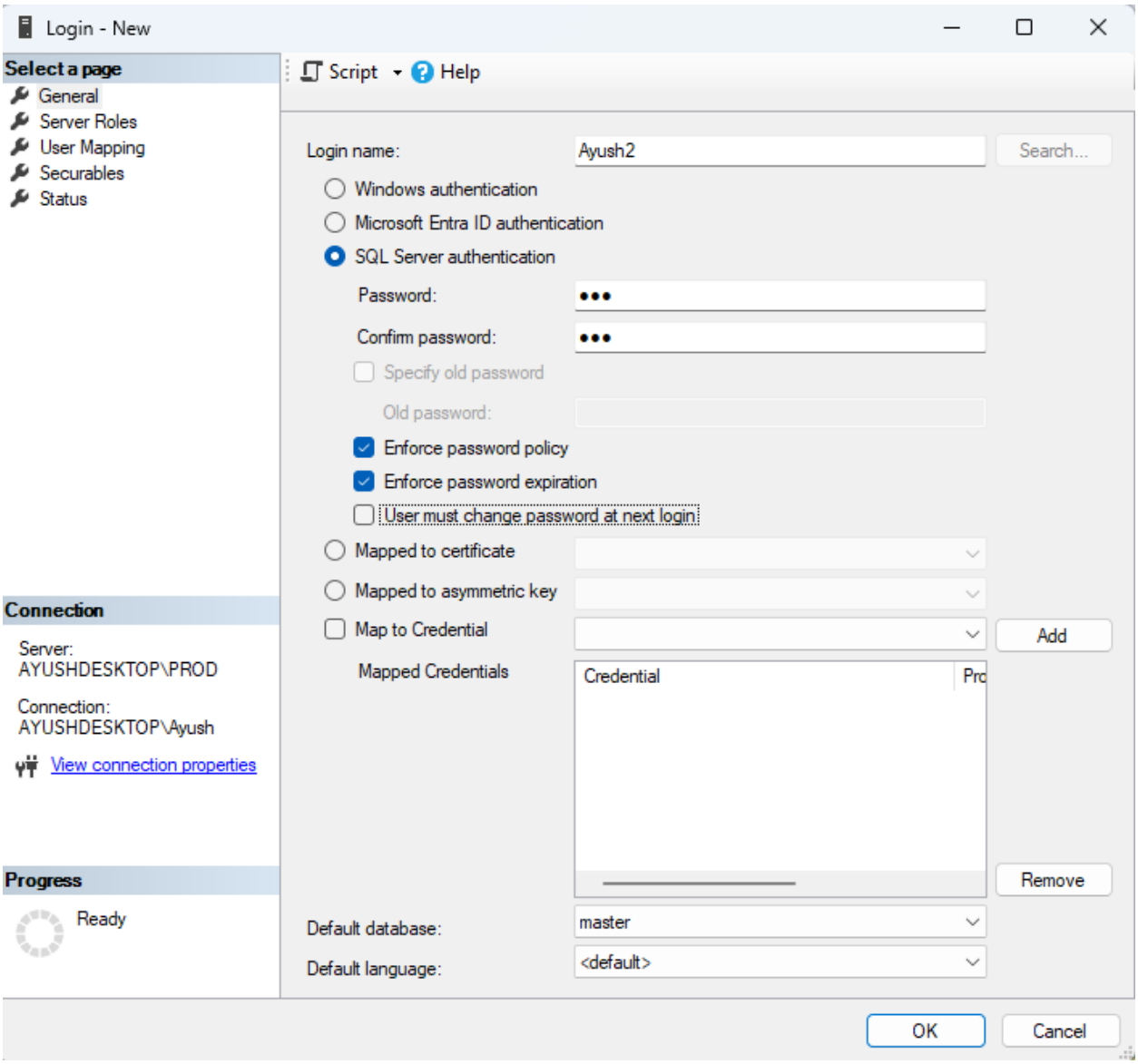
- Select New Login




Give, Login name and the necessary permissions.


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Ayush
Search...
Windows authentication
Microsoft Entra ID authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Pro
Remove
Connection
Server:
AYUSHDESKTOP\PROD
Connection:
AYUSHDESKTOP\Ayush
View connection properties
Progress
Ready
Default database:
master
Default language:
<default>
OK
Cancel




## ❖ Important Note

In SQL Server, when you create a new login, it is automatically assigned to the public server role. The public role is a special default role to which all logins belong. It serves as the baseline security level for every user in the database system. Here's what it entails:

### Permissions

- **Default Permissions**: The public role comes with a minimal set of permissions that allows users to connect to the server and its databases, but

with very restricted access. For example, members of the public can see the existence of objects within the database but may not have the right to read from or write to them unless explicitly granted.

- **Cannot Be Changed**: You cannot alter membership in the public role, nor can you change the permissions assigned to the public directly. Instead, you manage access by granting or denying specific permissions to the public role on individual objects within the database.

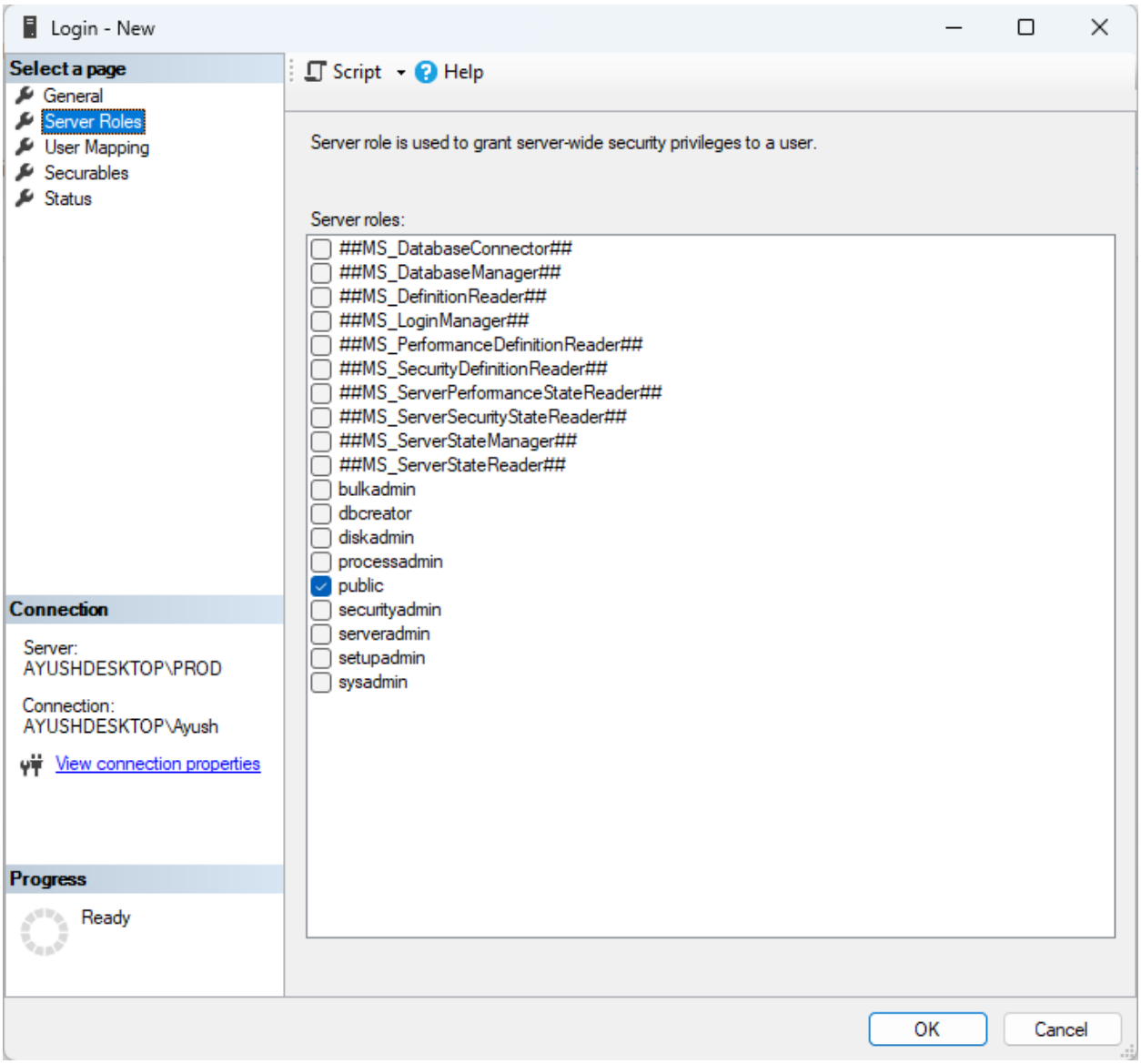
### Security Implications

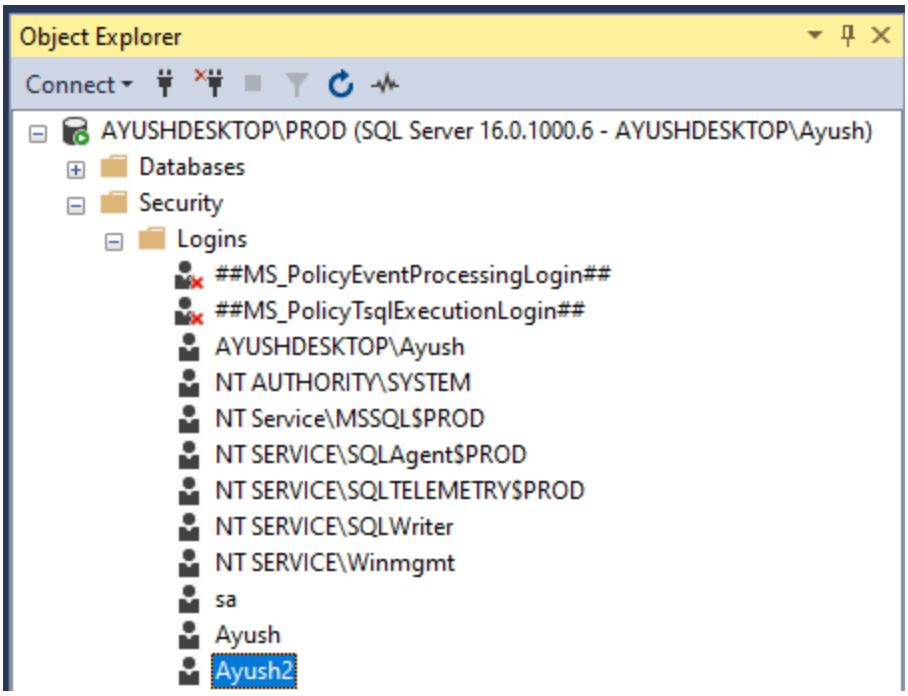
- **Initial Access**: Any new login will have the default access provided by the public role, ensuring that no login has more permissions than necessary unless specifically granted by a database administrator.
- **Role of public in Security**: This role acts as a security measure, ensuring that new users do not inadvertently receive broad access. It is up to the database administrators to elevate permissions based on the user's role and necessity.

### Best Practices

- **Use Specific Roles and Permissions**: It's best practice to use roles and explicitly defined permissions to manage what each user can and cannot do. Depending on the needs, administrators should create more specific roles and assign users to these roles, rather than relying on the public role for permission management.
- **Audit and Manage Permissions**: Regularly auditing who has what permissions through roles like public is important for maintaining security integrity. Ensuring that unnecessary permissions are not granted to the public can help minimize potential security risks.

The concept of public access in SQL Server is foundational for maintaining a secure and well-ordered database environment, especially in systems where multiple users interact with various databases and servers.


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Server role is used to grant server-wide security privileges to a user.
Server roles:
##MS_DatabaseConnector##
##MS_DatabaseManager##
##MS_DefinitionReader##
##MS_LoginManager##
##MS_PerformanceDefinitionReader##
##MS_SecurityDefinitionReader##
##MS_ServerPerformanceStateReader##
##MS_ServerSecurityStateReader##
##MS_ServerStateManager##
##MS_ServerStateReader##
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connection
Server:
AYUSHDESKTOP\PROD
Connection:
AYUSHDESKTOP\Ayush
View connection properties
Progress
Ready
OK
Cancel




## ❖ Important Note

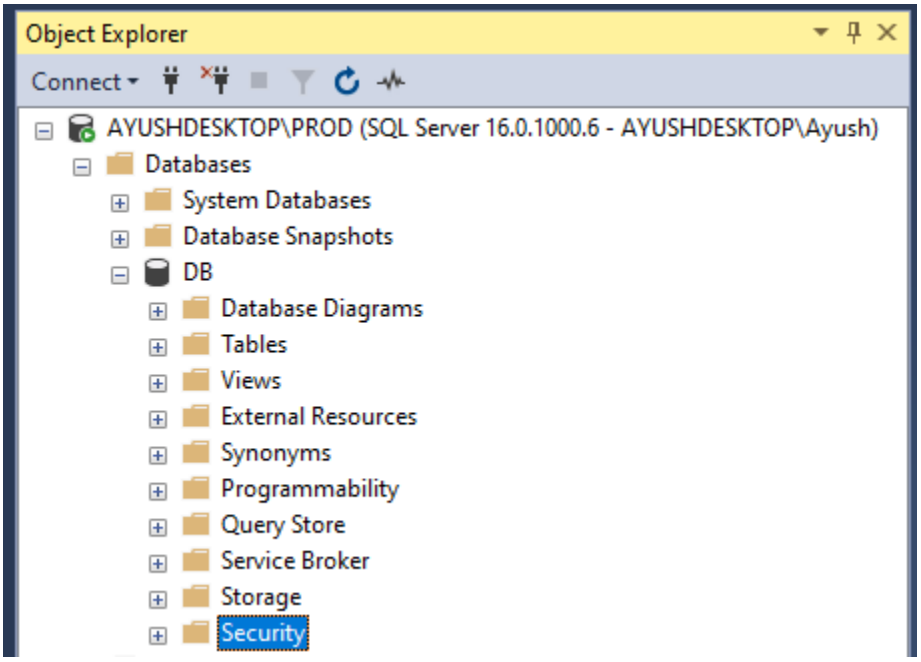
Creating users in a SQL Server database serves several important purposes:

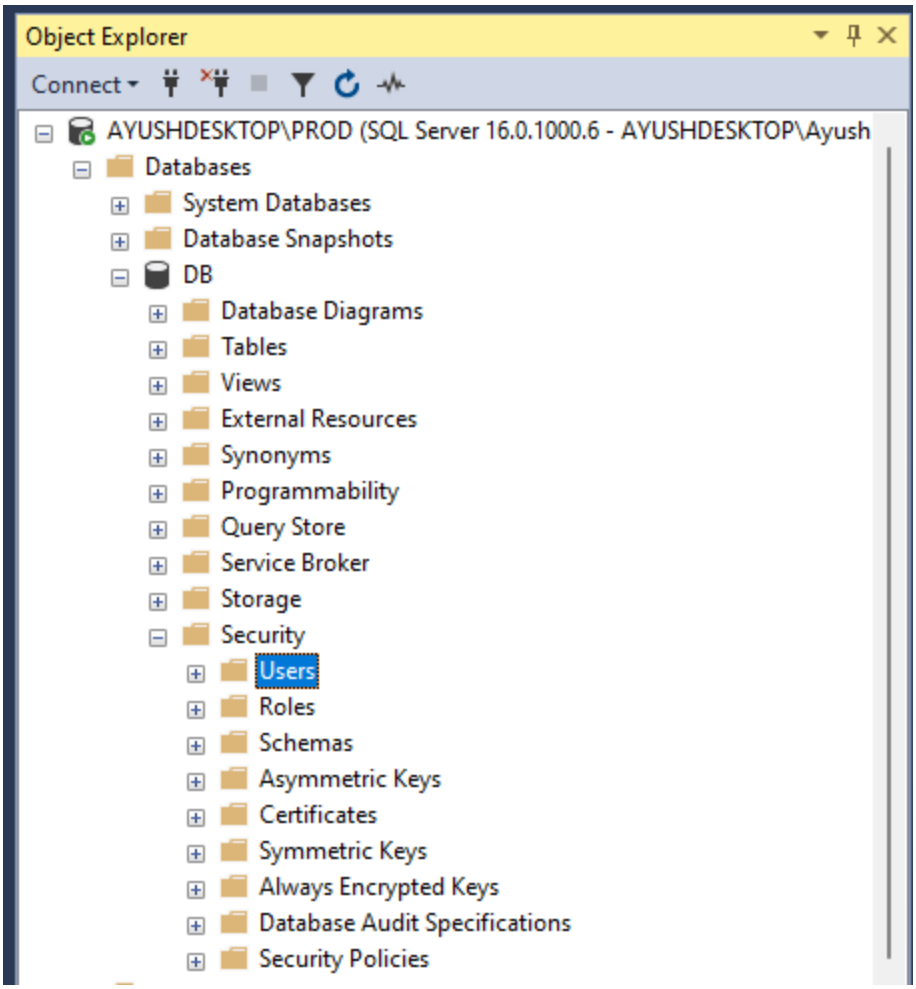
1. **Security Management**: Users help in managing access to database resources. By creating specific user accounts, you can enforce security measures, ensuring that only authorized individuals have access to sensitive data and operations.
2. **Role-Based Access Control**: Users can be assigned roles that define their permissions. This simplifies the management of privileges, as you can manage permissions at the role level rather than at the individual user level.
3. **Auditing and Compliance**: By having distinct user accounts, you can monitor and audit user activities more effectively. This is essential for compliance with various regulations and standards (like GDPR, HIPAA, etc.).
4. **Separation of Duties**: Creating different users allows for a clear separation of duties among database administrators, developers, and end-users, minimizing the risk of errors and fraud.
5. **Customization of Permissions**: Each user can have specific permissions tailored to their role or function. This granularity allows for better

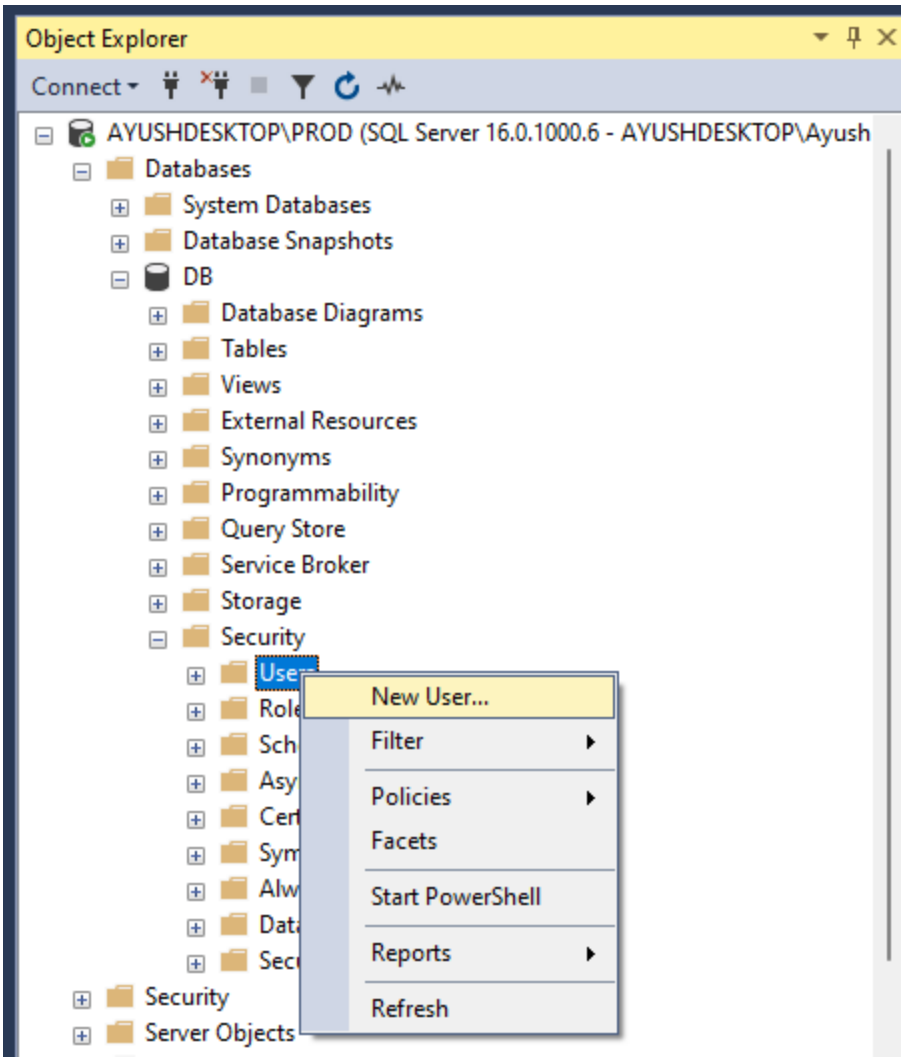
management of who can read, write, or execute certain actions in the database.

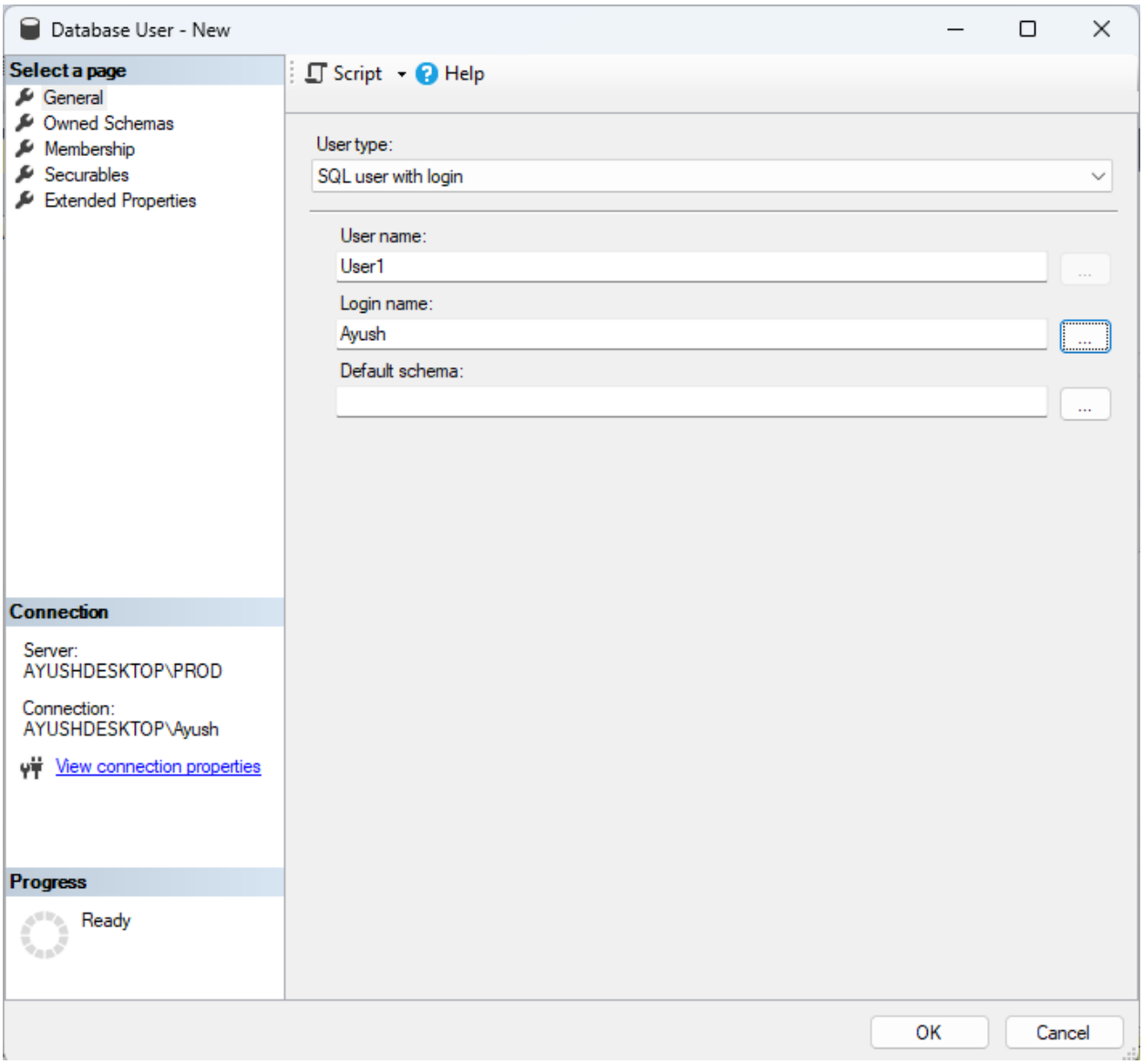
6. **Management of Database Resources**: Users can be created for different applications or services that interact with the database, allowing for more efficient resource management and troubleshooting.
7. **Improved Collaboration**: In environments where multiple people or teams are accessing the same database, having distinct user accounts helps facilitate collaboration while maintaining security.

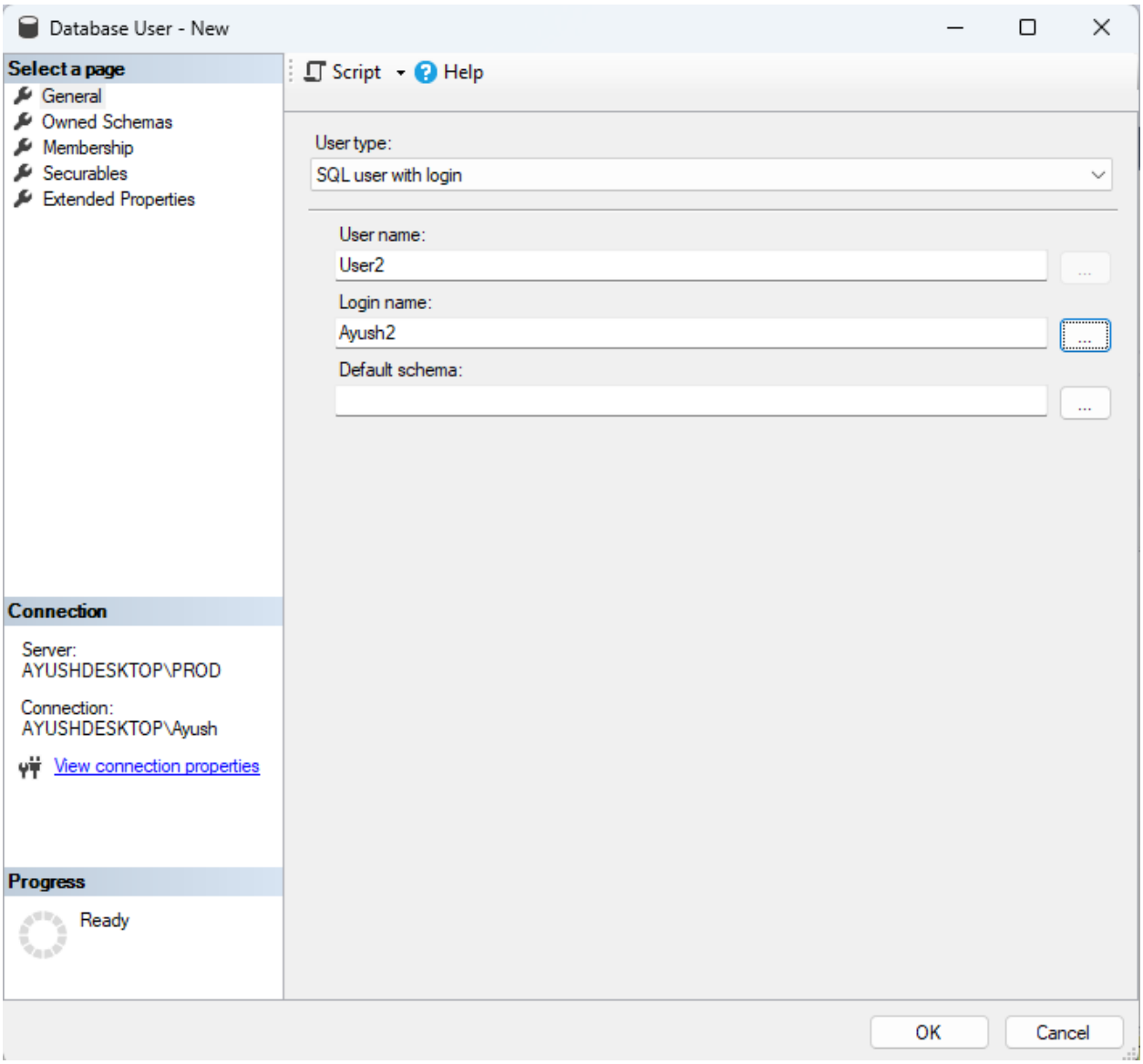
By implementing a structured approach to user management, you can enhance the security, efficiency, and integrity of your database systems.

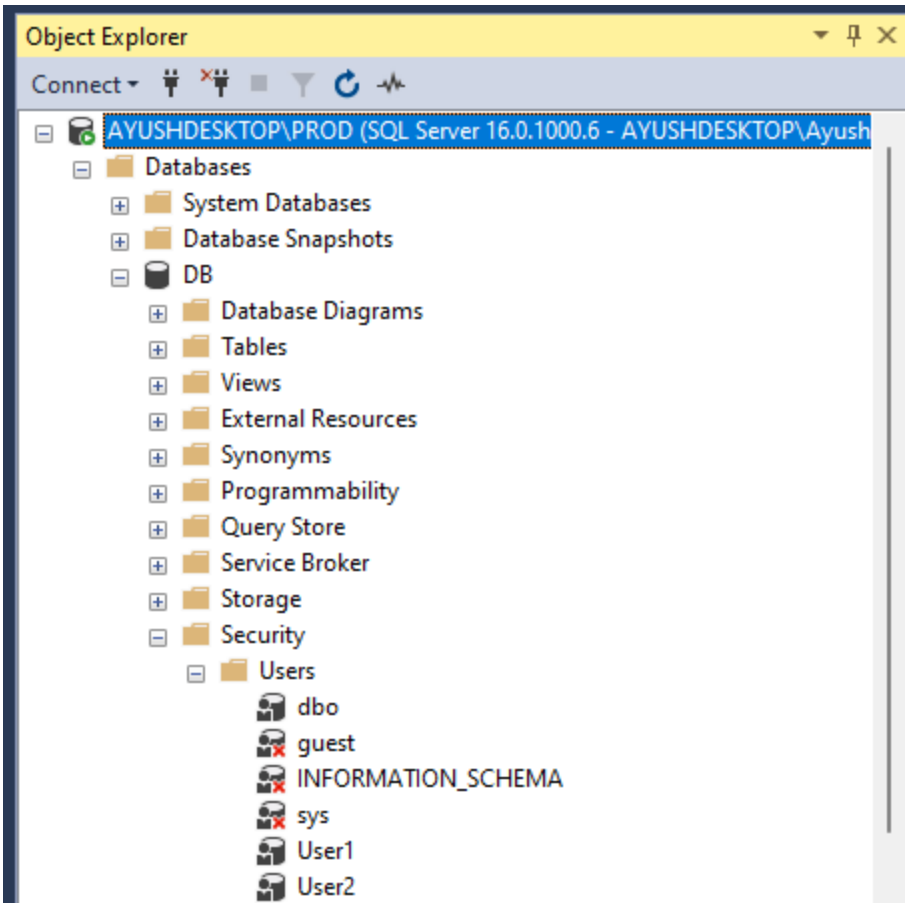


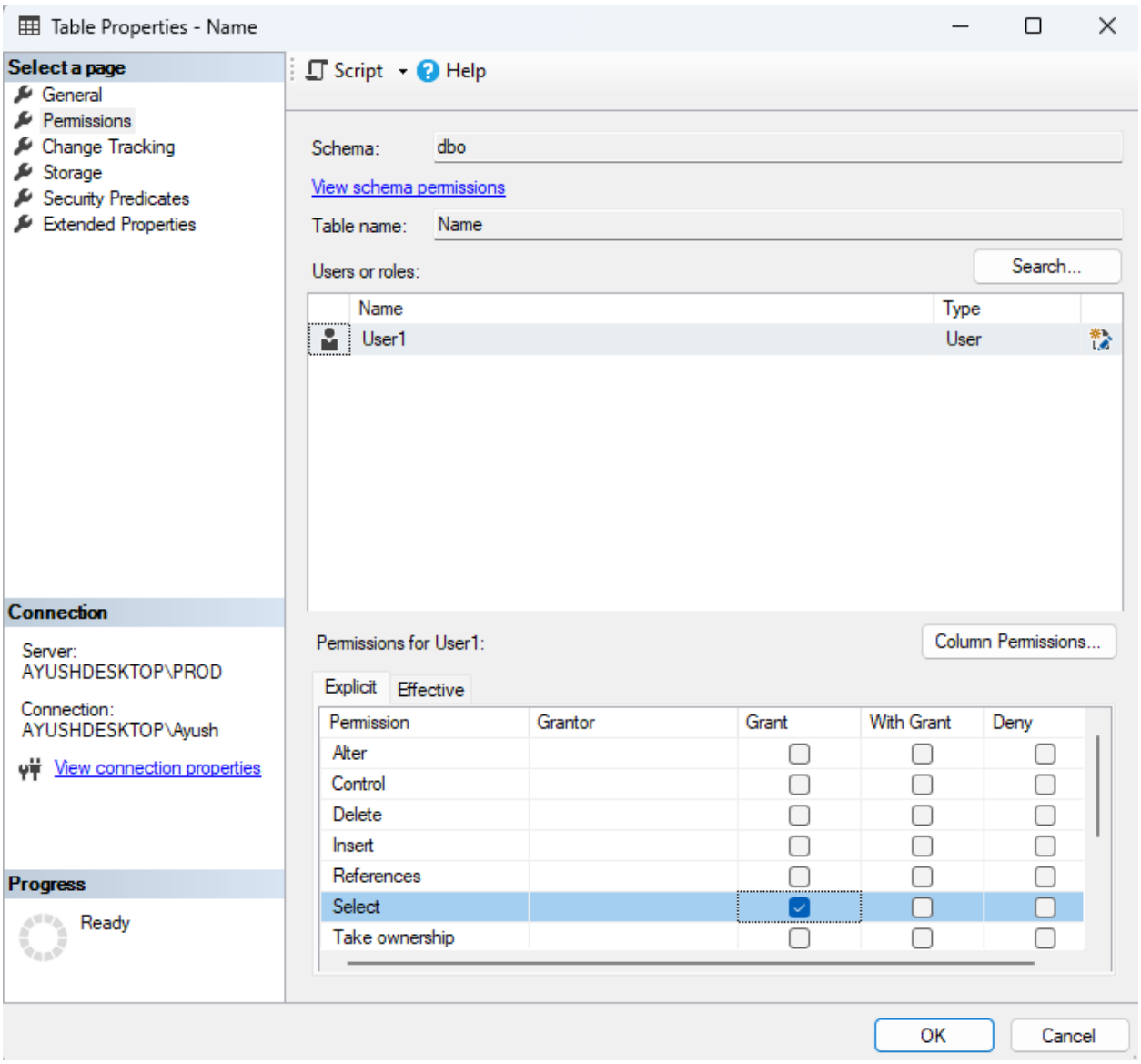

Object Explorer
Connect
AYUSHDESKTOP\PROD (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush
Databases
System Databases
Database Snapshots
DB
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Users
Roles
Schemas
Asymmetric Keys
Certificates
Symmetric Keys
Always Encrypted Keys
Database Audit Specifications
Security Policies


Object Explorer
Connect
AYUSHDESKTOP\PROD (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush
Databases
System Databases
Database Snapshots
DB
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
New User...
Filter
Policies
Facets
Start PowerShell
Reports
Refresh
Security
Server Objects


Database User - New
Select a page
General
Owned Schemas
Membership
Securables
Extended Properties
Script
Help
User type:
SQL user with login
User name:
User1
Login name:
Ayush
Default schema:
Connection
Server:
AYUSHDESKTOP\PROD
Connection:
AYUSHDESKTOP\Ayush
View connection properties
Progress
Ready
OK
Cancel


Database User - New
Select a page
General
Owned Schemas
Membership
Securables
Extended Properties
Script
Help
User type:
SQL user with login
User name:
User2
Login name:
Ayush2
Default schema:
Connection
Server:
AYUSHDESKTOP\PROD
Connection:
AYUSHDESKTOP\Ayush
View connection properties
Progress
Ready
OK
Cancel


Object Explorer
Connect
AYUSHDESKTOP\PROD (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush
Databases
System Databases
Database Snapshots
DB
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Users
dbo
guest
INFORMATION_SCHEMA
sys
User1
User2




User1 doesn’t have the Insert permission on the Name table

SQLQuery2.sql - AY...OD.DB (Ayush (64))*

```sql
Use DB

Select * From Name

Insert into Name values('Haroon')
```

100 %

Messages

```
Msg 229, Level 14, State 5, Line 6
The INSERT permission was denied on the object 'Name', database 'DB', schema 'dbo'.

Completion time: 2024-09-25T17:23:14.1877076+05:30
```

SQLQuery3.sql - AY...D.DB (Ayush2 (69))*

```sql
--For User2 has every permission

Use DB

Select * From Name

Insert into Name values('Sidharth')
```

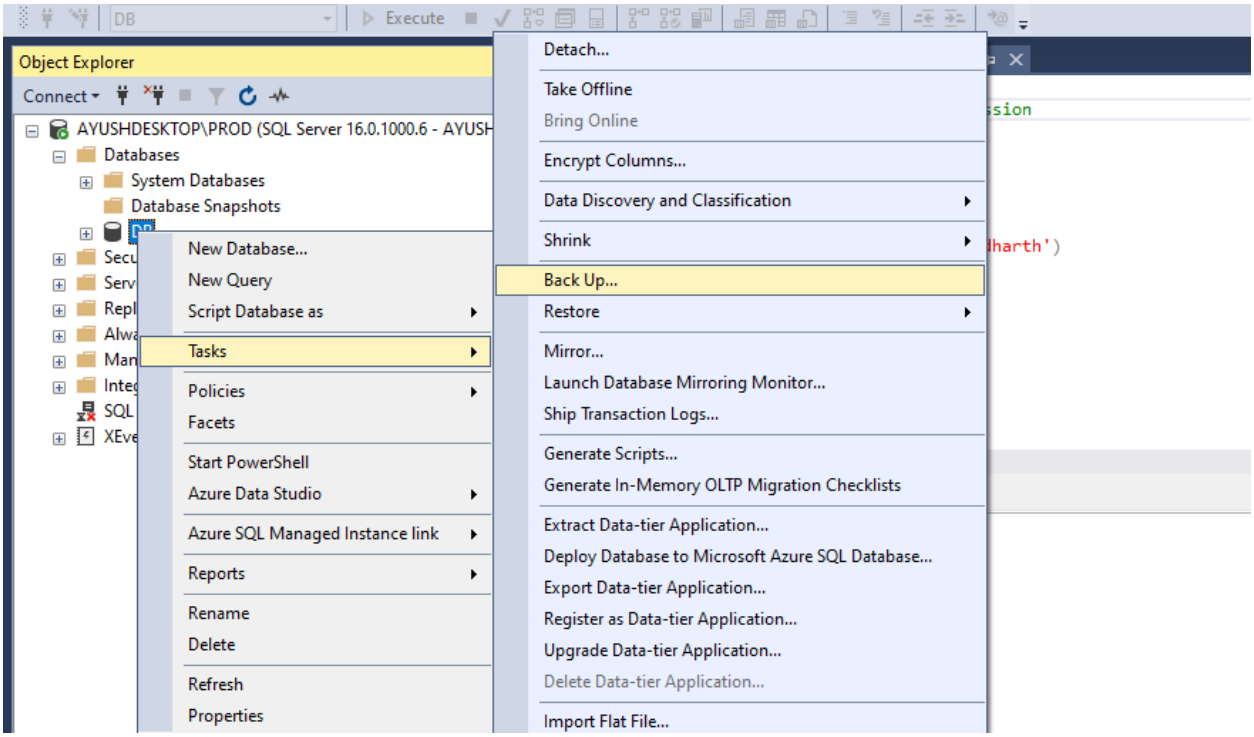
100 %

Results | Messages

| | Name |
|---|---|
| 1 | Shweta |
| 2 | Ravi |
| 3 | Manish |
| 4 | Wasim |
| 5 | Sidharth |

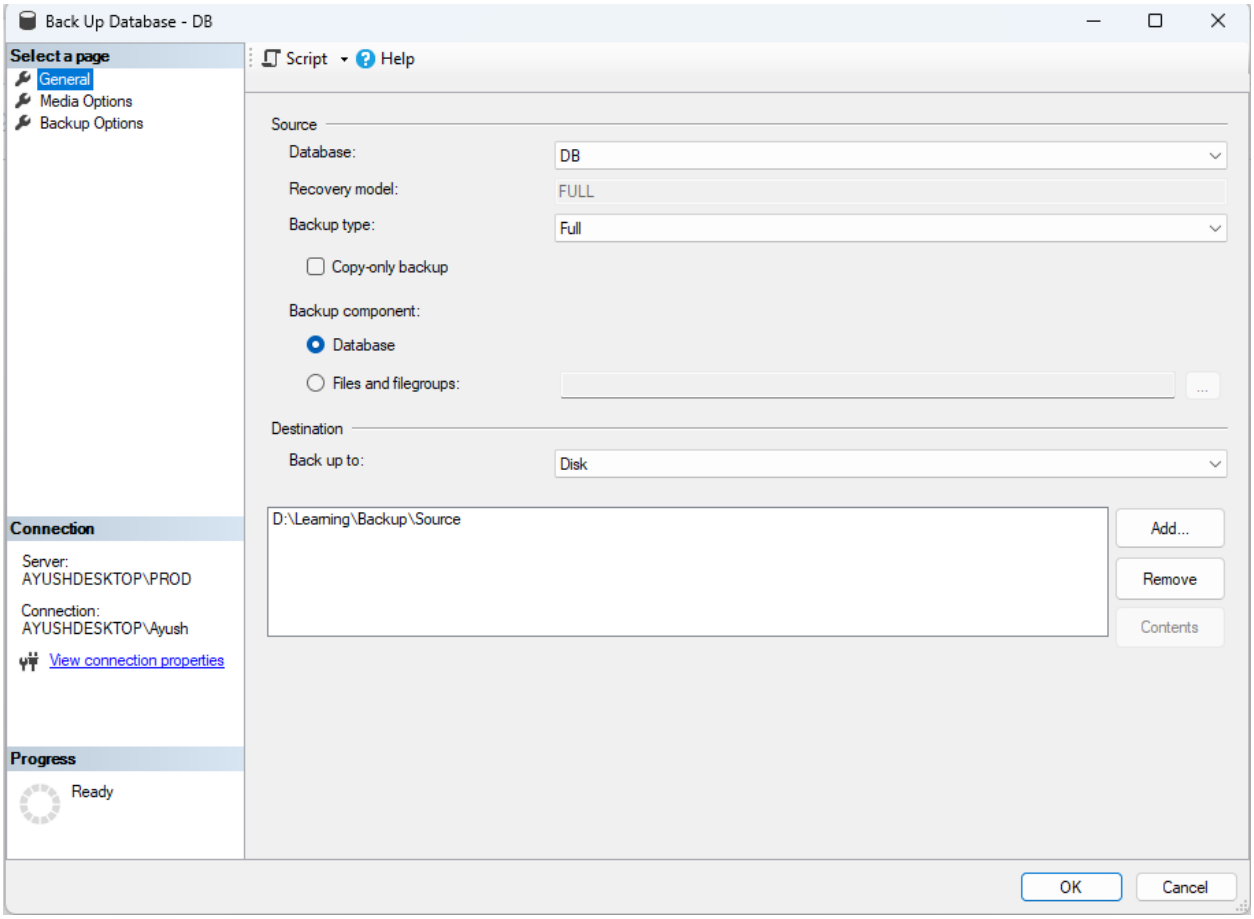
# Database Backup and Restoration

- **Backup Process: Utilized SQL Server Management Studio to create backups of the database, specifying compression and other parameters to optimize the backup operation.**



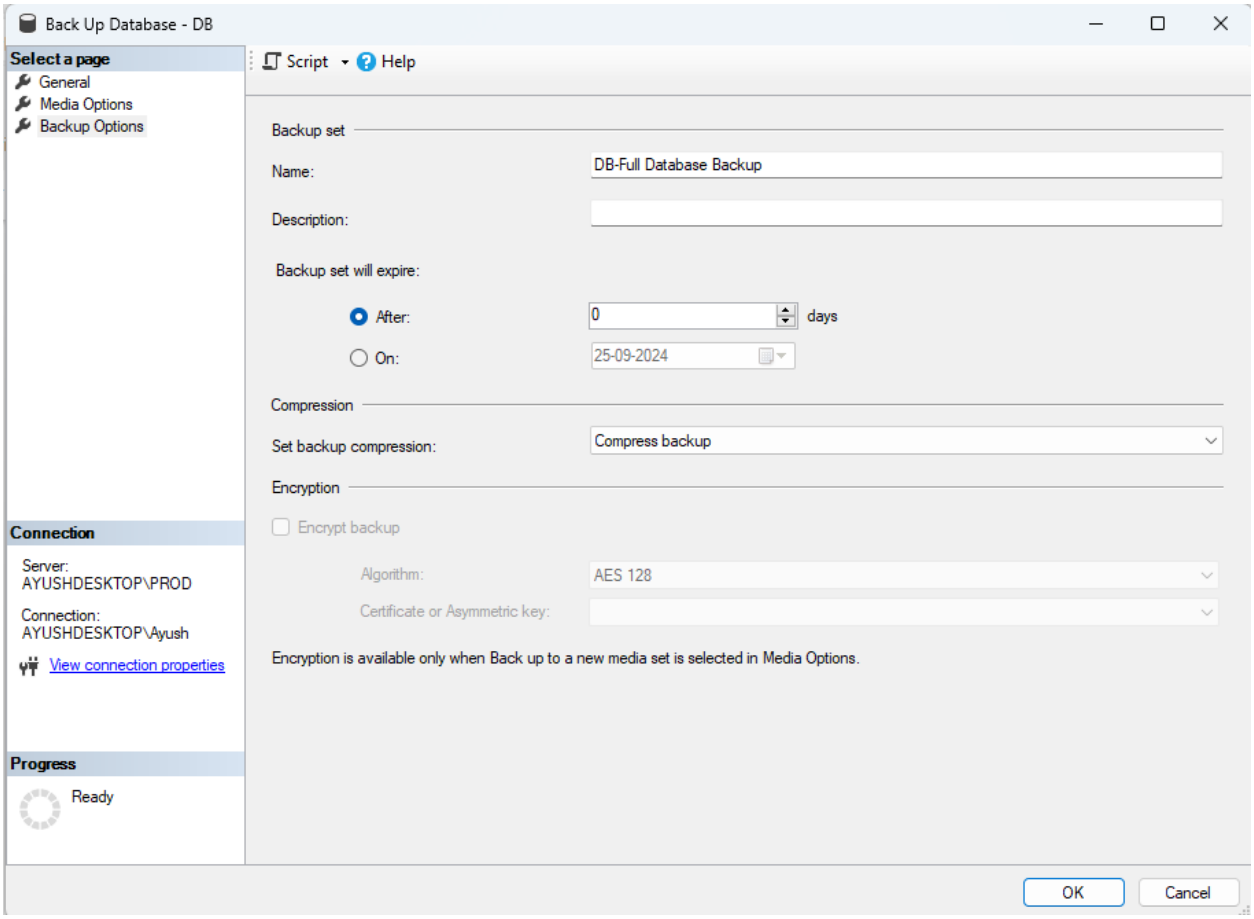

## ❖ Important Note

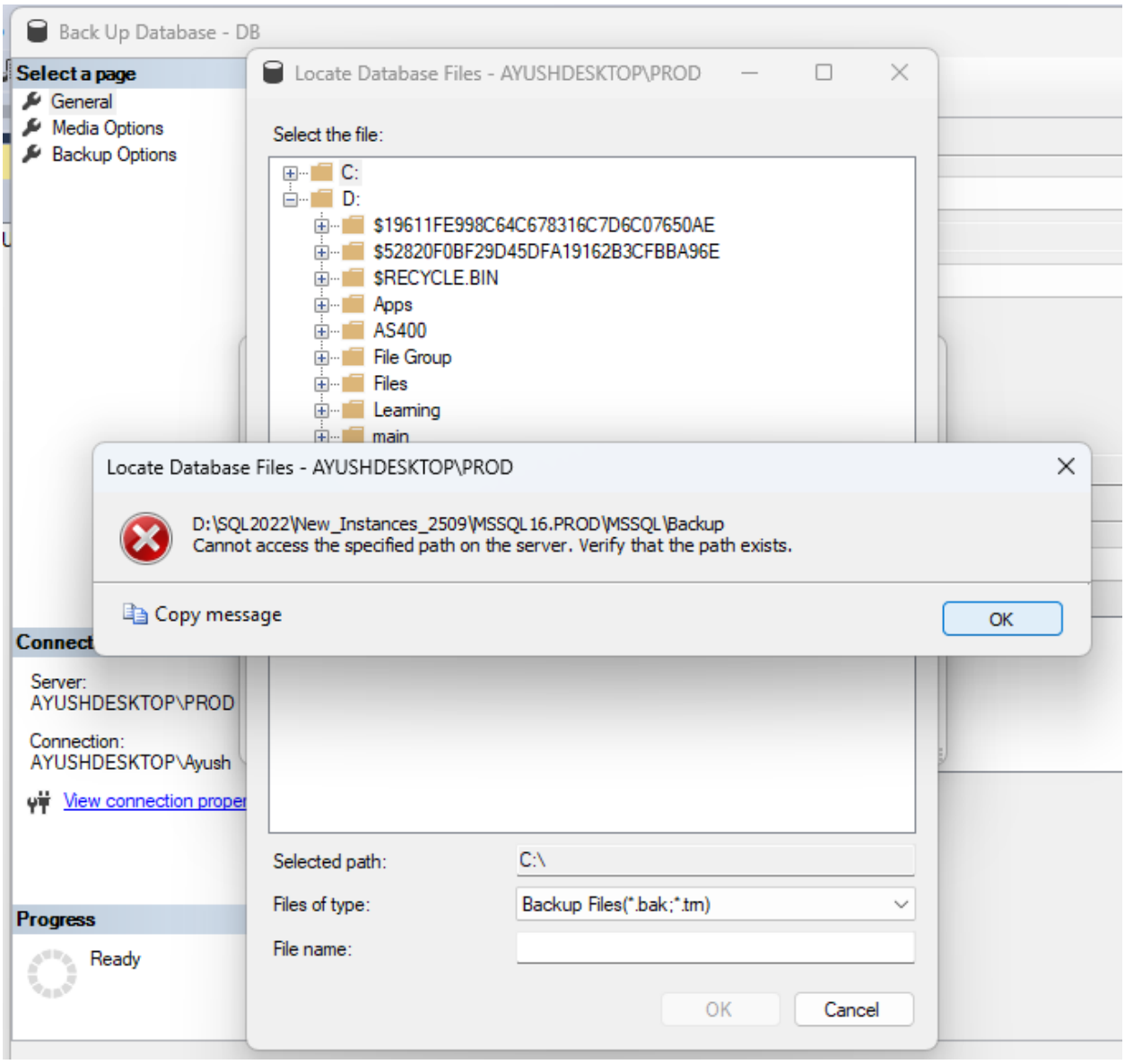
In SQL Server, the **full backup option** refers to a specific type of backup operation that creates a complete and comprehensive backup of an entire database at a specific point in time.




## ❖ Important Note

In SQL Server, the **compressed backup option** allows you to create backups that take up less disk space than standard backups.

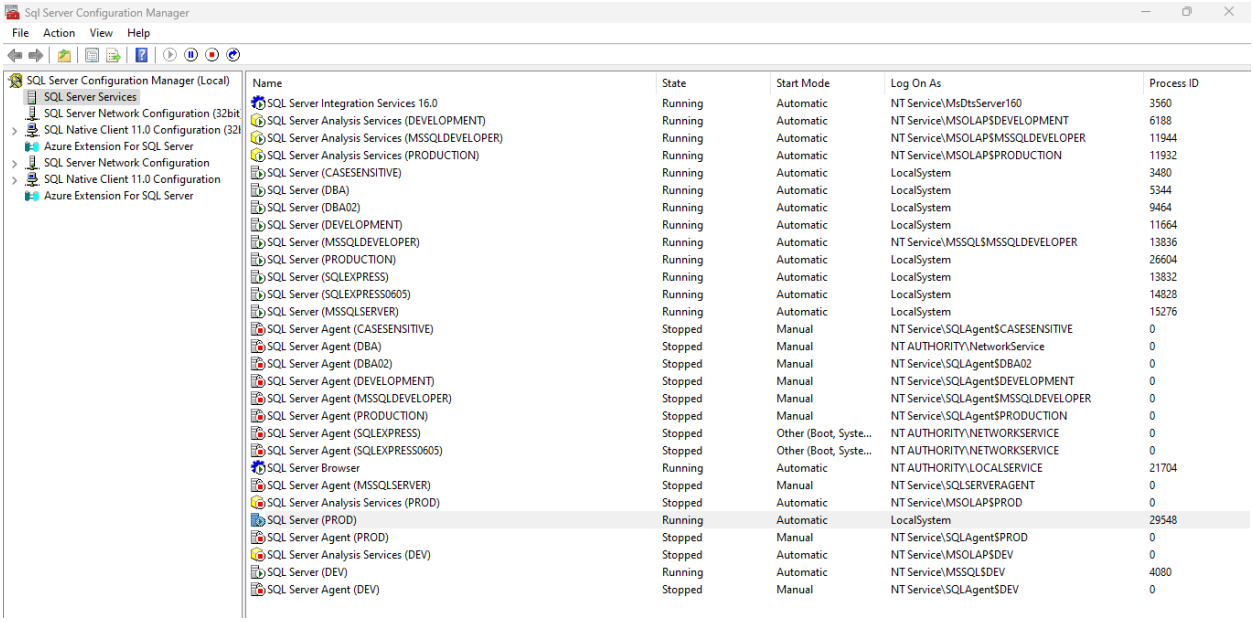





## Challenge 1: Database File Location Error

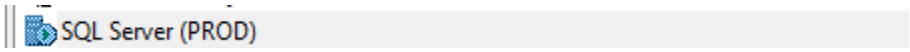
- **Problem:** Received errors regarding inaccessible database file paths.
- **Solution:** Modified the SQL Server service properties to run under a local system account that possesses necessary file system permissions.

❑ Open SQL Server Configuration Manager

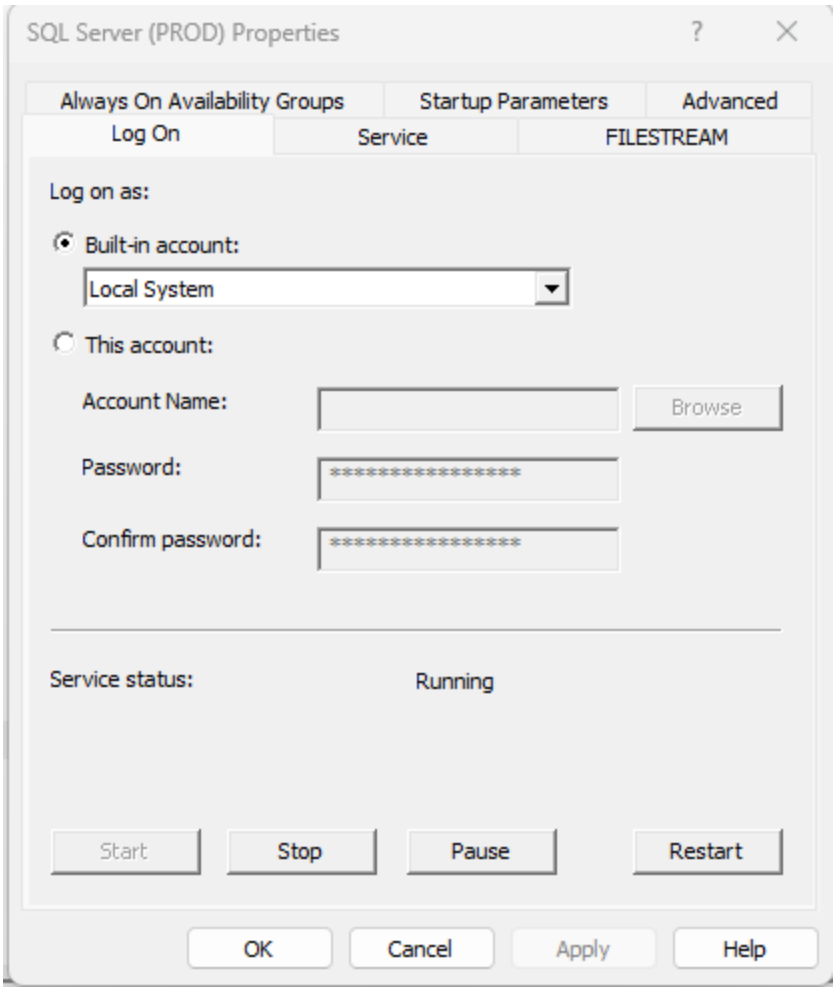


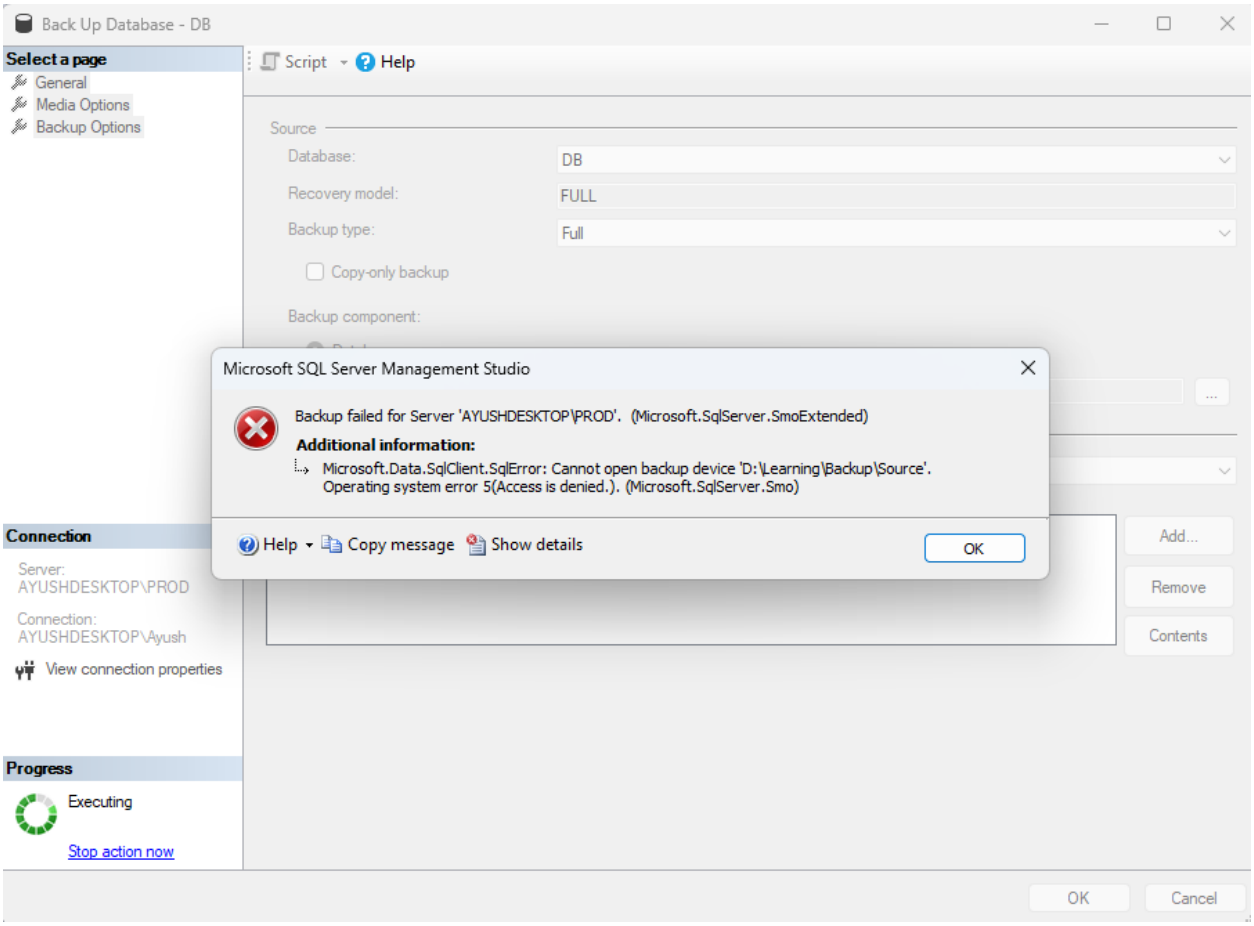

| Name | State | Start Mode | Log On As | Process ID |
|---|---|---|---|---|
| SQL Server Integration Services 16.0 | Running | Automatic | NT Service\MsDtsServer160 | 3560 |
| SQL Server Analysis Services (DEVELOPMENT) | Running | Automatic | NT Service\MSOLAP$DEVELOPMENT | 6188 |
| SQL Server Analysis Services (MSSQLDEVELOPER) | Running | Automatic | NT Service\MSOLAP$MSSQLDEVELOPER | 11944 |
| SQL Server Analysis Services (PRODUCTION) | Running | Automatic | NT Service\MSOLAP$PRODUCTION | 11932 |
| SQL Server (CASESENSITIVE) | Running | Automatic | LocalSystem | 3480 |
| SQL Server (DBA) | Running | Automatic | LocalSystem | 5344 |
| SQL Server (DBA02) | Running | Automatic | LocalSystem | 9464 |
| SQL Server (DEVELOPMENT) | Running | Automatic | LocalSystem | 11664 |
| SQL Server (MSSQLDEVELOPER) | Running | Automatic | NT Service\MSSQL$MSSQLDEVELOPER | 13836 |
| SQL Server (PRODUCTION) | Running | Automatic | LocalSystem | 26604 |
| SQL Server (SQLEXPRESS) | Running | Automatic | LocalSystem | 13832 |
| SQL Server (SQLEXPRESS0605) | Running | Automatic | LocalSystem | 14828 |
| SQL Server (MSSQLSERVER) | Running | Automatic | LocalSystem | 15276 |
| SQL Server Agent (CASESENSITIVE) | Stopped | Manual | NT Service\SQLAgent$CASESENSITIVE | 0 |
| SQL Server Agent (DBA) | Stopped | Manual | NT AUTHORITY\NetworkService | 0 |
| SQL Server Agent (DBA02) | Stopped | Manual | NT Service\SQLAgent$DBA02 | 0 |
| SQL Server Agent (DEVELOPMENT) | Stopped | Manual | NT Service\SQLAgent$DEVELOPMENT | 0 |
| SQL Server Agent (MSSQLDEVELOPER) | Stopped | Manual | NT Service\SQLAgent$MSSQLDEVELOPER | 0 |
| SQL Server Agent (PRODUCTION) | Stopped | Manual | NT Service\SQLAgent$PRODUCTION | 0 |
| SQL Server Agent (SQLEXPRESS) | Stopped | Other (Boot, Syste... | NT AUTHORITY\NETWORKSERVICE | 0 |
| SQL Server Agent (SQLEXPRESS0605) | Stopped | Other (Boot, Syste... | NT AUTHORITY\NETWORKSERVICE | 0 |
| SQL Server Browser | Running | Automatic | NT AUTHORITY\LOCALSERVICE | 21704 |
| SQL Server Agent (MSSQLSERVER) | Stopped | Manual | NT Service\SQLSERVERAGENT | 0 |
| SQL Server Analysis Services (PROD) | Stopped | Automatic | NT Service\MSOLAP$PROD | 0 |
| SQL Server (PROD) | Running | Automatic | LocalSystem | 29548 |
| SQL Server Agent (PROD) | Stopped | Manual | NT Service\SQLAgent$PROD | 0 |
| SQL Server Analysis Services (DEV) | Stopped | Automatic | NT Service\MSOLAP$DEV | 0 |
| SQL Server (DEV) | Running | Automatic | NT Service\MSSQL$DEV | 4080 |
| SQL Server Agent (DEV) | Stopped | Manual | NT Service\SQLAgent$DEV | 0 |

- ❑ SQL Server Configuration Manager (Local) - SQL Server Services
- ❑ Select Instance name




- ❑ Right Click – Properties – Select – Built – in – account - Select Local System
- ❑ Apply - Ok

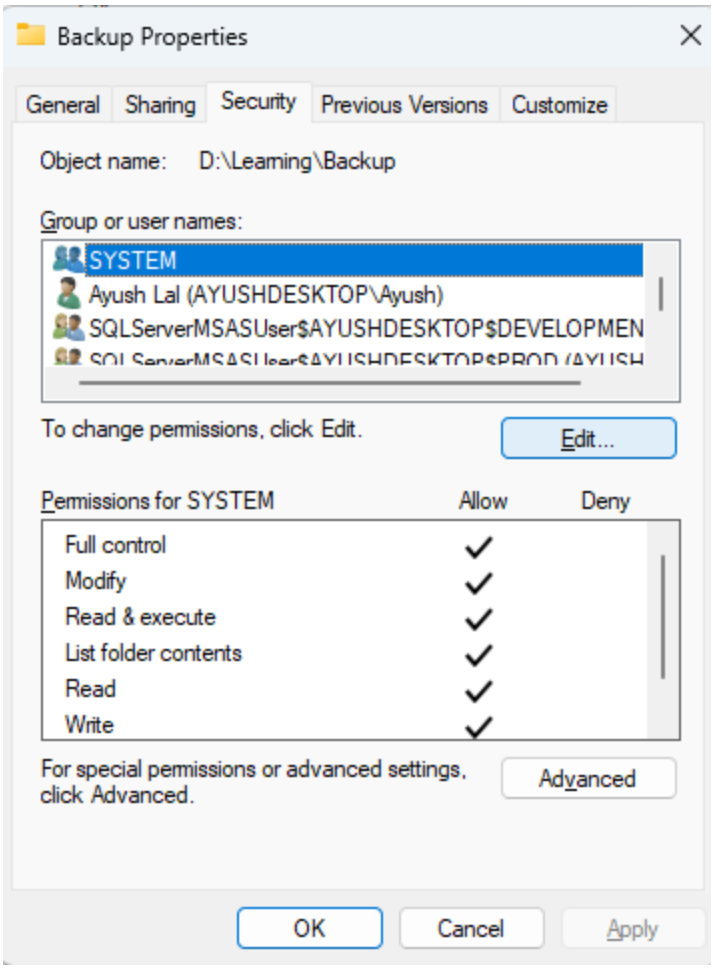

SQL Server (PROD) Properties
Always On Availability Groups
Startup Parameters
Advanced
Log On
Service
FILESTREAM
Log on as:
Built-in account:
Local System
This account:
Account Name:
Browse
Password:
Confirm password:
Service status:
Running
Start
Stop
Pause
Restart
OK
Cancel
Apply
Help



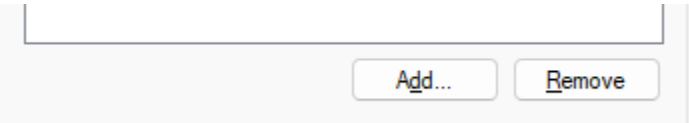

## Challenge 2: Access Denied Error

- **Problem:** Encountered an 'Access is denied' error while attempting to back up the database.
- **Solution:** Adjusted folder permissions to allow the SQL Server service account access to the backup directory, ensuring the service had full control.

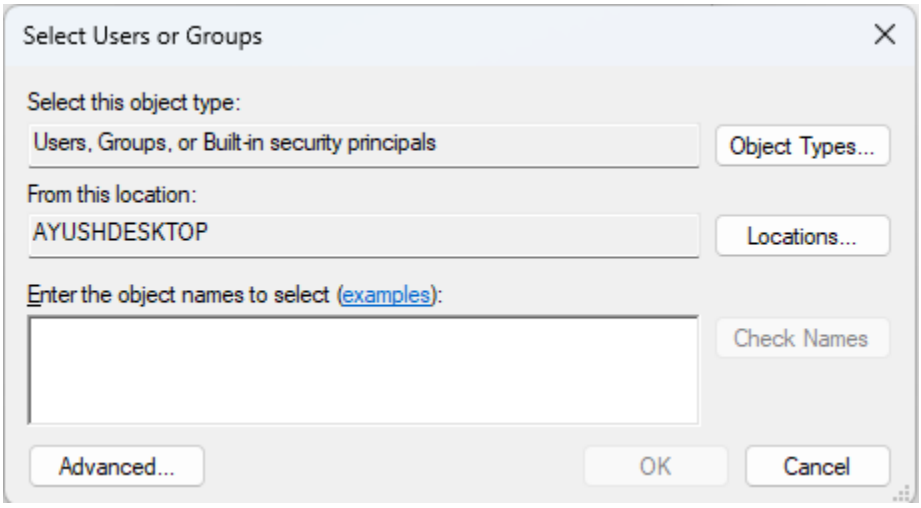
❑ Right Click Backup Folder – Properties – Security Tab – Click Edit



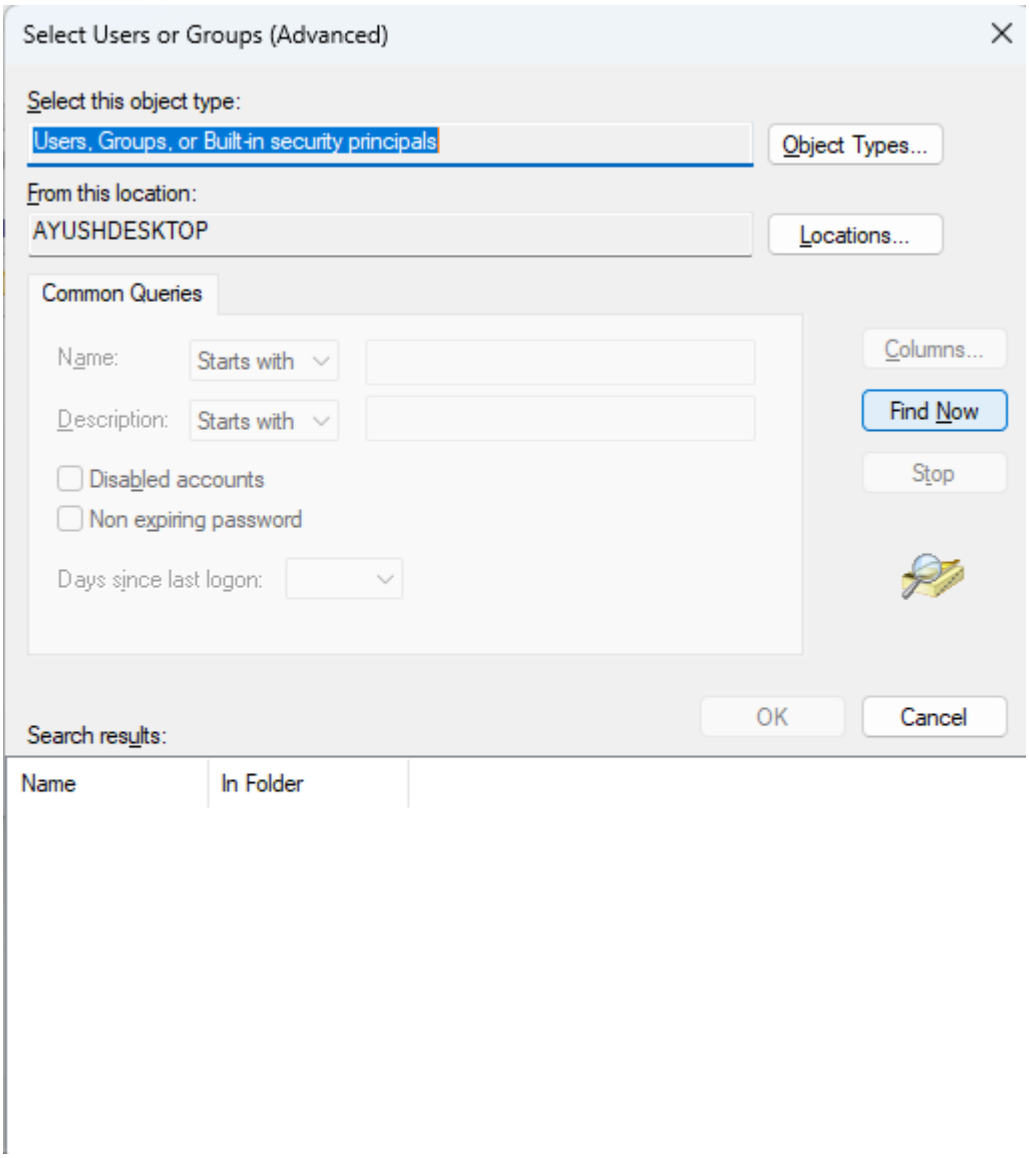

- ❑ Click Add



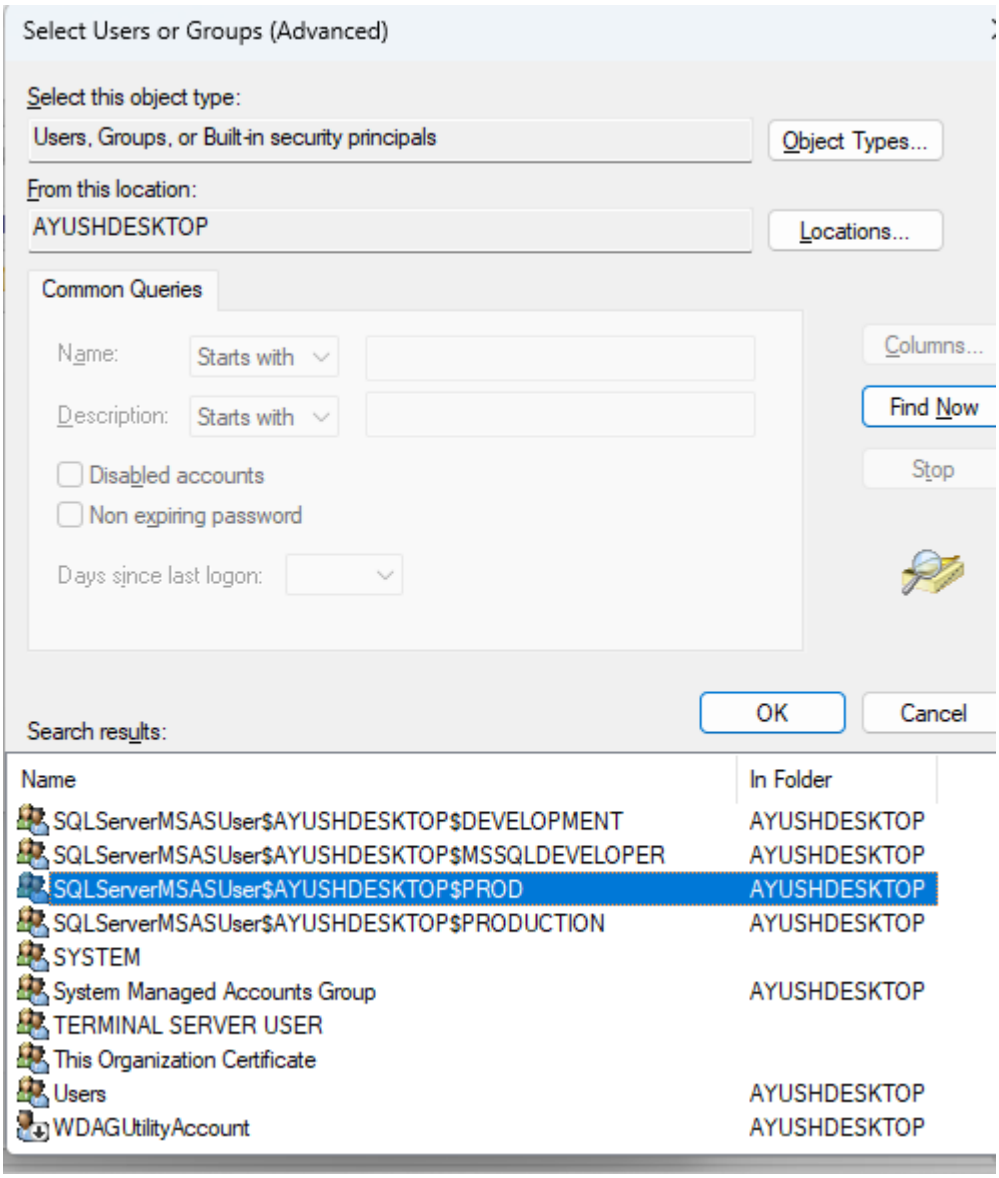

- ❑ Select Advanced




- ❑ Select Find Now

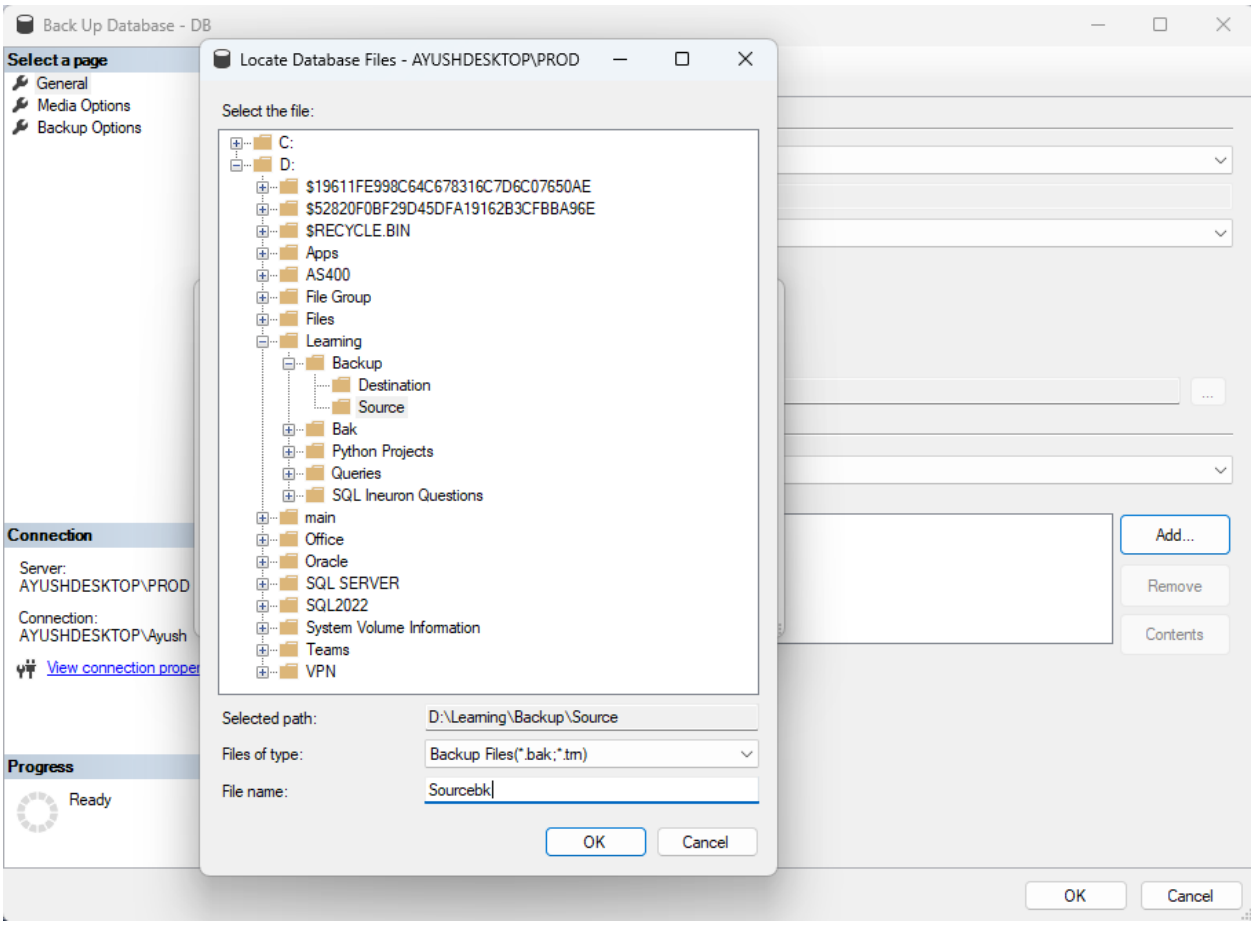


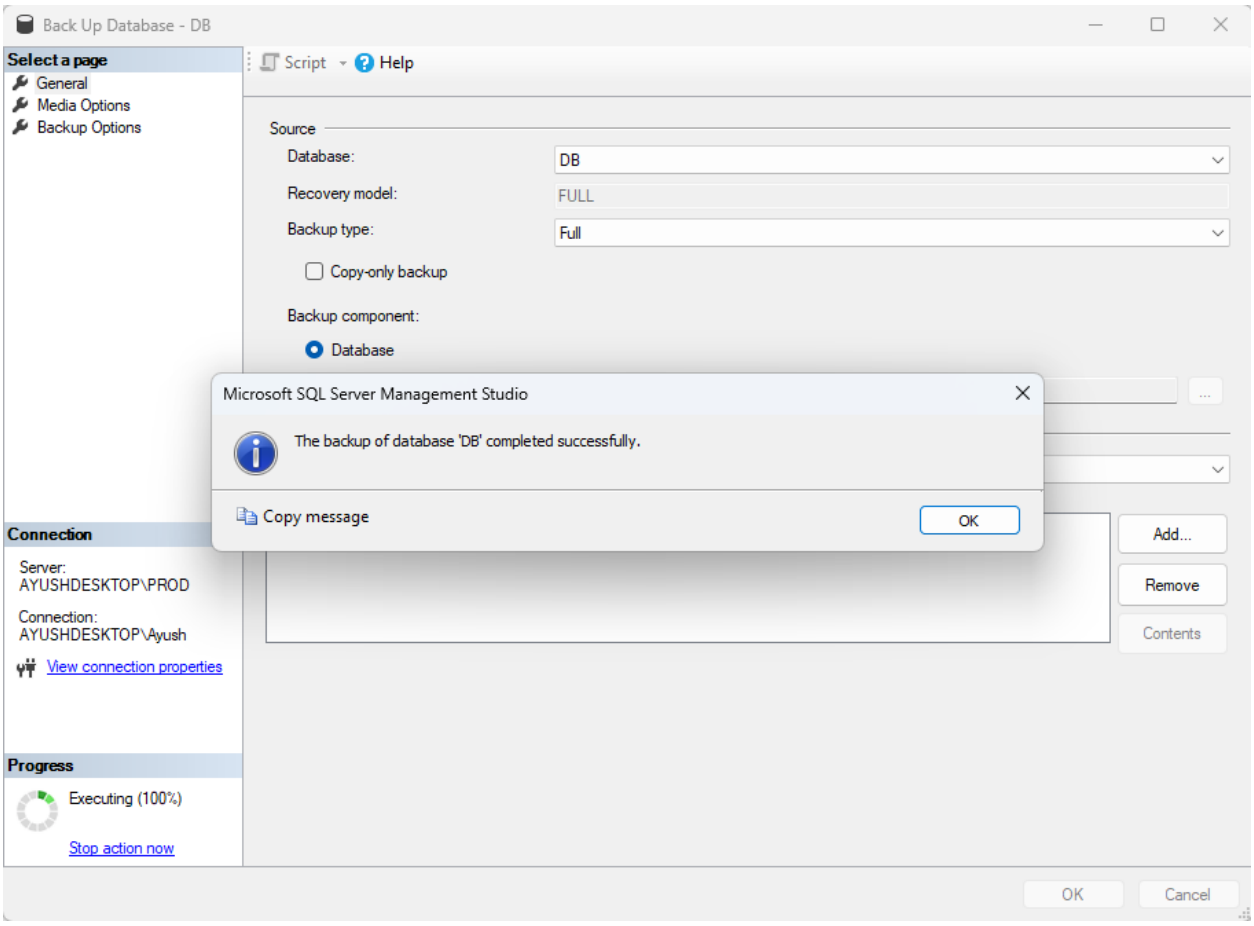

- Select Instance Name:




- ❑ Ok – Ok – Select Full Control – Apply – Ok

- ❑ Proceed with the remaining Backup Procedures


Back Up Database - DB
Select a page
General
Media Options
Backup Options
Connection
Server:
AYUSHDESKTOP\PROD
Connection:
AYUSHDESKTOP\Ayush
View connection proper
Progress
Ready
Locate Database Files - AYUSHDESKTOP\PROD
Select the file:
C:
D:
$19611FE998C64C678316C7D6C07650AE
$52820F0BF29D45DFA19162B3CFBBA96E
$RECYCLE.BIN
Apps
AS400
File Group
Files
Learning
Backup
Destination
Source
Bak
Python Projects
Queries
SQL Ineuron Questions
main
Office
Oracle
SQL SERVER
SQL2022
System Volume Information
Teams
VPN
Selected path:
D:\Learning\Backup\Source
Files of type:
Backup Files(*.bak;*.tm)
File name:
Sourcebk
OK
Cancel
Add...
Remove
Contents
OK
Cancel




## SCRIPT FOR BACKUP

```
BACKUP DATABASE [DB] TO  DISK =
N'D:\Learning\Backup\Source\Sourcebk' WITH NOFORMAT, NOINIT,

NAME = N'DB-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,
COMPRESSION,  STATS = 10

GO
```

SQLQuery4.sql - A...ESKTOP\Ayush (58))*

```sql
BACKUP DATABASE [DB] TO  DISK = N'D:\Learning\Backup\Source\Sourcebk' WITH NOFORMAT, NOINIT,
NAME = N'DB-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, COMPRESSION,  STATS = 10
GO
```

100 %

Messages

```
11 percent processed.
21 percent processed.
31 percent processed.
41 percent processed.
51 percent processed.
61 percent processed.
71 percent processed.
81 percent processed.
91 percent processed.
100 percent processed.
Processed 576 pages for database 'DB', file 'DB' on file 1.
Processed 1 pages for database 'DB', file 'DB_log' on file 1.
BACKUP DATABASE successfully processed 577 pages in 0.030 seconds (150.048 MB/sec).

Completion time: 2024-09-25T17:51:32.5955501+05:30
```

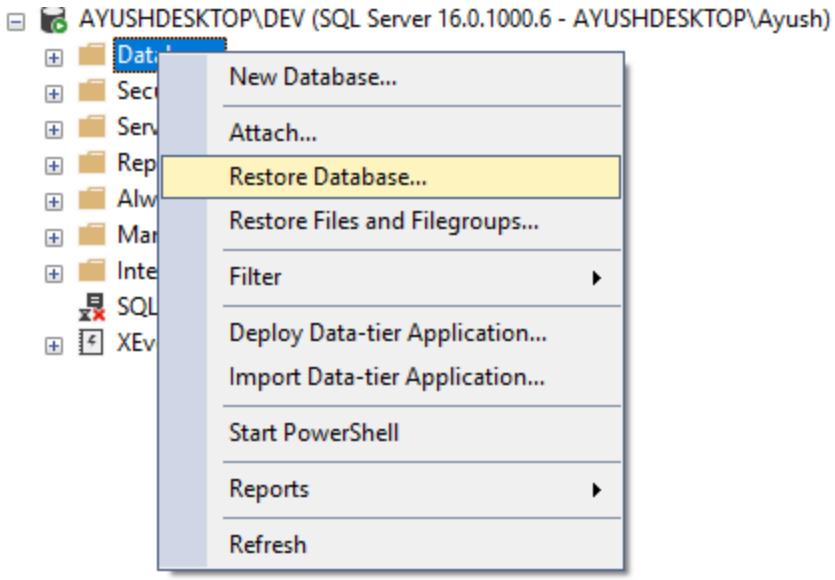
Backup Taken

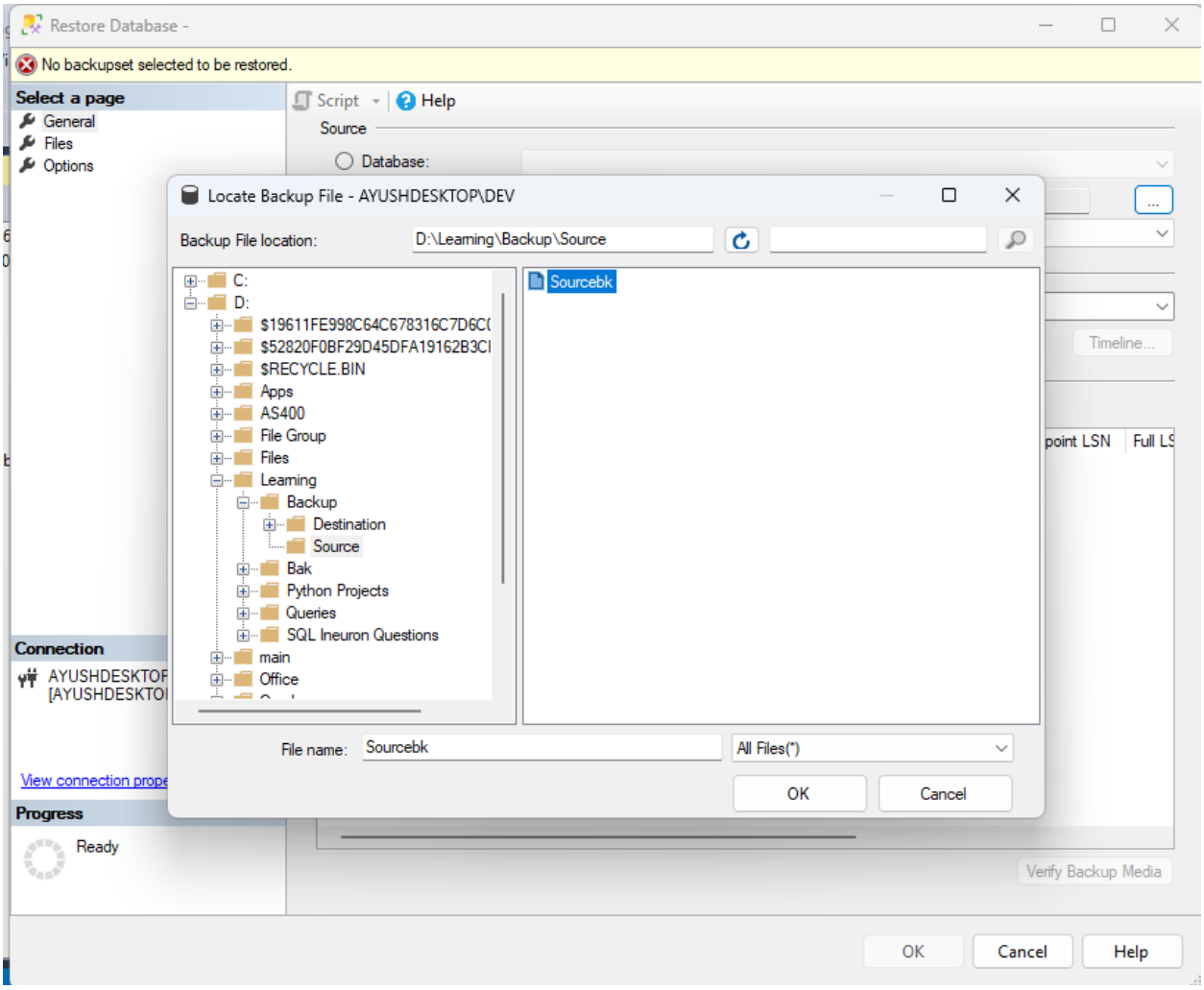
This PC > Local Disk (D:) > Learning > Backup > Source

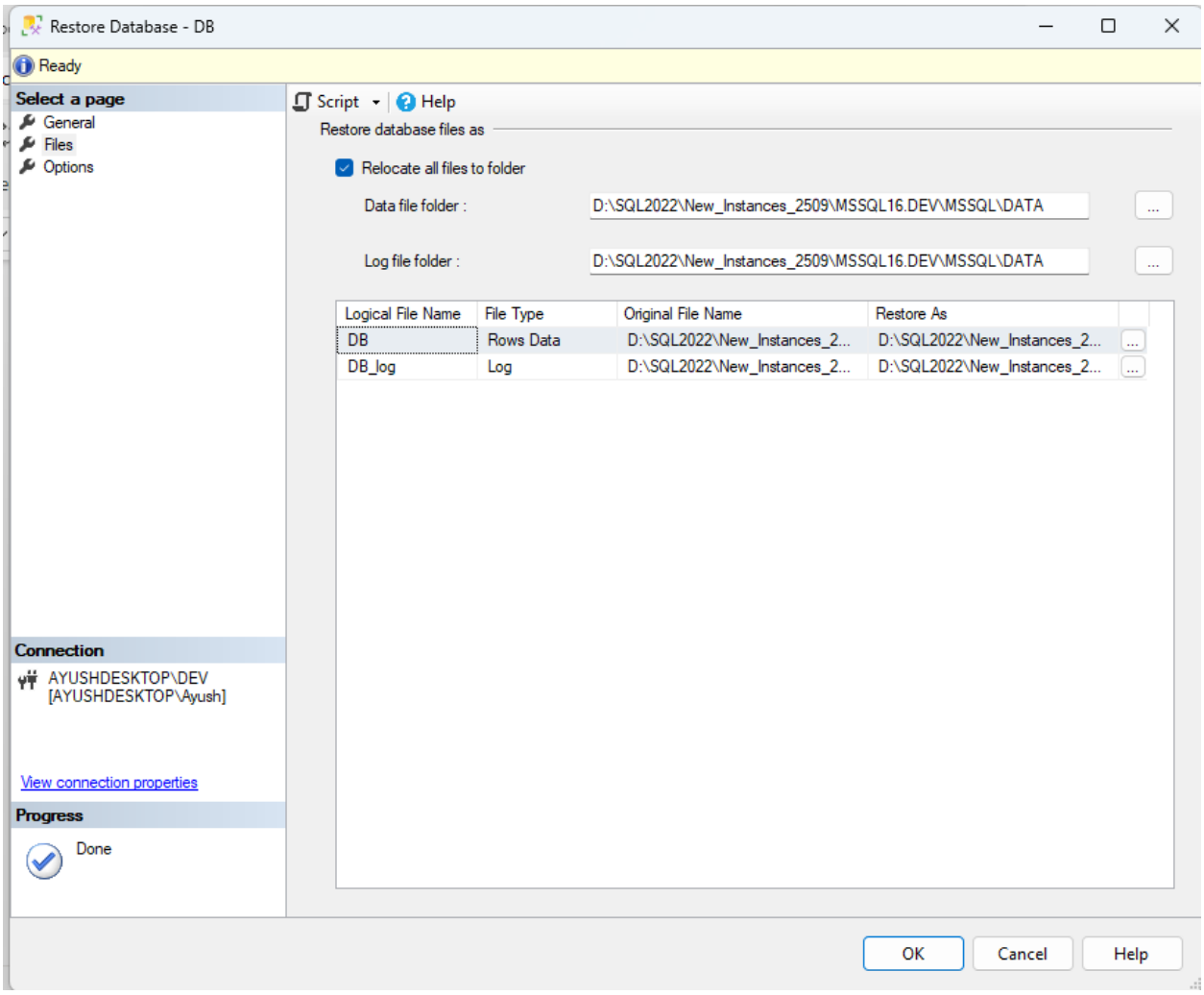
Sort View ...

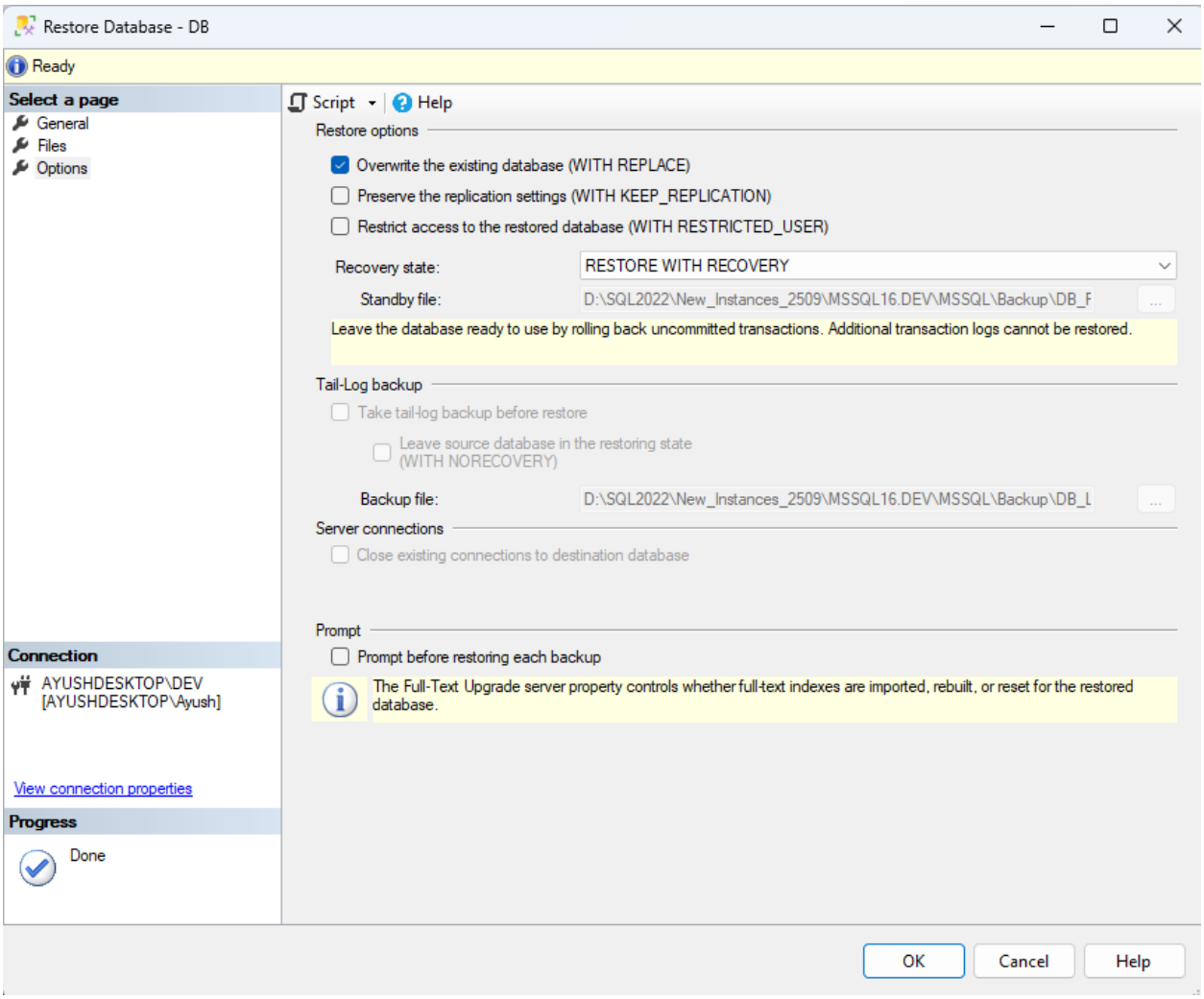
| Name | Date modified | Type | Size |
|---|---|---|---|
| Sourcebk | 25-09-2024 17:51 | File | 613 KB |

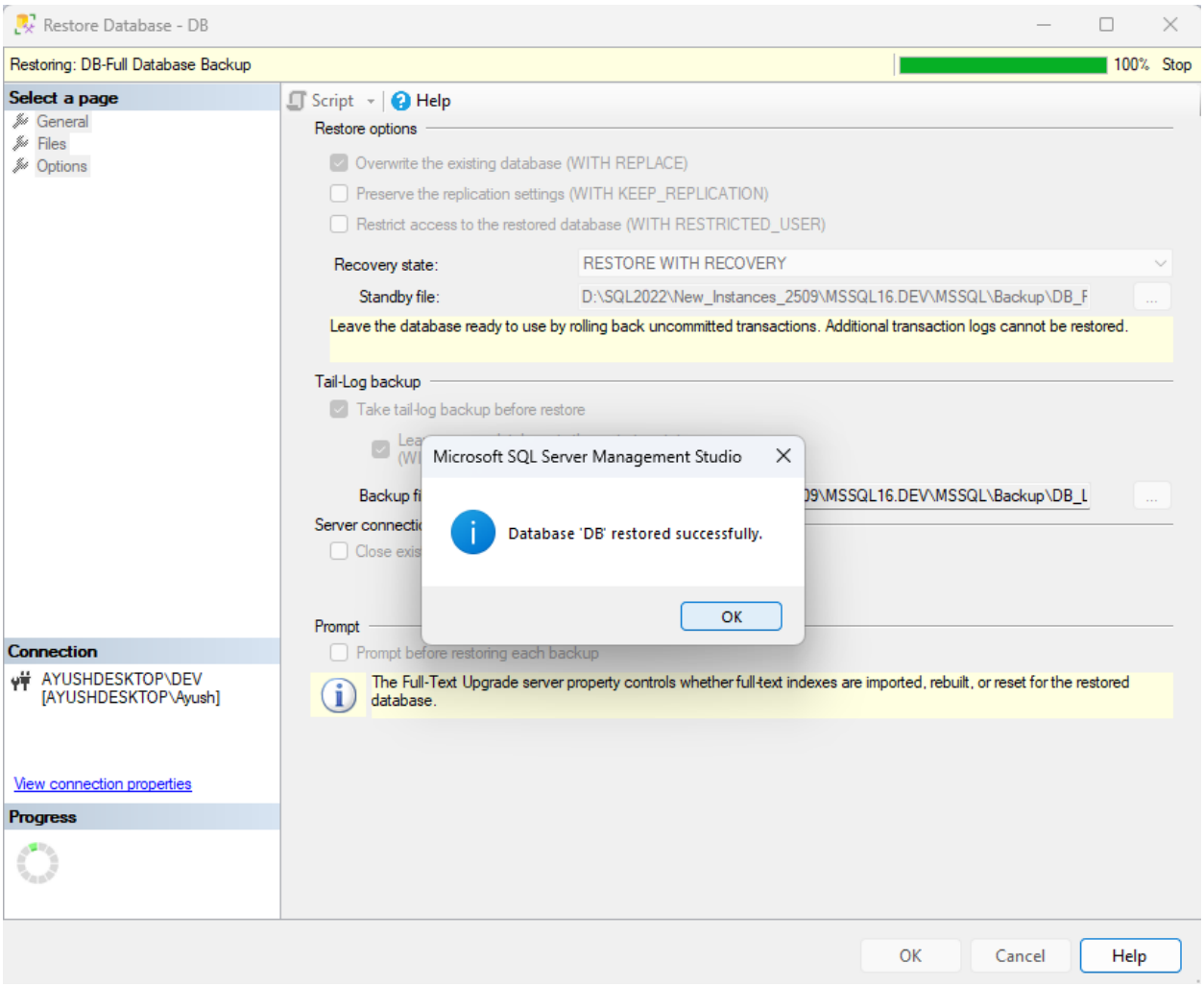
- **Restore Process:** Restored the backup to the DEV instance, adjusting file paths and ensuring environmental consistency.


AYUSHDESKTOP\DEV (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush)
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh


Restore Database -
No backupset selected to be restored.
Select a page
General
Files
Options
Script
Help
Source
Database:
Locate Backup File - AYUSHDESKTOP\DEV
Backup File location:
D:\Learning\Backup\Source
C:
D:
$19611FE998C64C678316C7D6C(
$52820F0BF29D45DFA19162B3CI
$RECYCLE.BIN
Apps
AS400
File Group
Files
Learning
Backup
Destination
Source
Bak
Python Projects
Queries
SQL Ineuron Questions
main
Office
Sourcebk
File name:
Sourcebk
All Files(*)
OK
Cancel
Timeline...
Connection
View connection prope
Progress
Ready
Verify Backup Media
OK
Cancel
Help


Restore Database - DB
Ready
Select a page
General
Files
Options
Script
Help
Restore database files as
Relocate all files to folder
Data file folder :
D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA
Log file folder :
D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA
Logical File Name
File Type
Original File Name
Restore As
DB
Rows Data
D:\SQL2022\New_Instances_2...
D:\SQL2022\New_Instances_2...
DB_log
Log
D:\SQL2022\New_Instances_2...
D:\SQL2022\New_Instances_2...
Connection
AYUSHDESKTOP\DEV
[AYUSHDESKTOP\Ayush]
View connection properties
Progress
Done
OK
Cancel
Help


Restore Database - DB
Ready
Select a page
General
Files
Options
Script
Help
Restore options
Overwrite the existing database (WITH REPLACE)
Preserve the replication settings (WITH KEEP_REPLICATION)
Restrict access to the restored database (WITH RESTRICTED_USER)
Recovery state:
RESTORE WITH RECOVERY
Standby file:
D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\Backup\DB_F
Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.
Tail-Log backup
Take tail-log backup before restore
Leave source database in the restoring state (WITH NORECOVERY)
Backup file:
D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\Backup\DB_L
Server connections
Close existing connections to destination database
Prompt
Prompt before restoring each backup
The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.
Connection
AYUSHDESKTOP\DEV
[AYUSHDESKTOP\Ayush]
View connection properties
Progress
Done
OK
Cancel
Help




**SCRIPT**

```
USE [master]

RESTORE DATABASE [DB] FROM  DISK =
N'D:\Learning\Backup\Source\Sourcebk' WITH  FILE = 1,

MOVE N'DB' TO
N'D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA\DB.mdf',

MOVE N'DB_log' TO
N'D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA\DB_log.ldf
',

NOUNLOAD,  REPLACE,  STATS = 5   GO
```

SQLQuery6.sql - A...ESKTOP\Ayush (77))*

```sql
USE [master]
RESTORE DATABASE [DB] FROM  DISK = N'D:\Learning\Backup\Source\Sourcebk' WITH  FILE = 1,
MOVE N'DB' TO N'D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA\DB.mdf',
MOVE N'DB_log' TO N'D:\SQL2022\New_Instances_2509\MSSQL16.DEV\MSSQL\DATA\DB_log.ldf',
NOUNLOAD,  REPLACE,  STATS = 5

GO
```

100 %

Messages

```
5 percent processed.
11 percent processed.
15 percent processed.
21 percent processed.
25 percent processed.
31 percent processed.
35 percent processed.
41 percent processed.
45 percent processed.
51 percent processed.
55 percent processed.
61 percent processed.
65 percent processed.
71 percent processed.
75 percent processed.
81 percent processed.
85 percent processed.
91 percent processed.
95 percent processed.
100 percent processed.
```

100 %

Object Explorer

Connect

- AYUSHDESKTOP\PROD (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush
- AYUSHDESKTOP\DEV (SQL Server 16.0.1000.6 - AYUSHDESKTOP\Ayush)
  - Databases
    - System Databases
    - Database Snapshots
    - DB
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Name
        - Dropped Ledger Tables

## Ensuring Security and Continuity

- **Login and User Sync:** Exported login details from the PROD server and imported them into the DEV server to maintain security settings.

Run this Script on PROD Environment

```
-- Script to extract logins from PROD
DECLARE @sql NVARCHAR(MAX) = '';
SELECT @sql += 'CREATE LOGIN [' + l.name + ']
WITH PASSWORD = ' + CONVERT(VARCHAR(MAX), l.password_hash, 1) + '
HASHED, SID = ' + CONVERT(VARCHAR(MAX), l.sid, 1) + ',
DEFAULT_DATABASE = [' + l.default_database_name + '];' + CHAR(13)
FROM sys.sql_logins l
WHERE l.is_disabled = 0;   PRINT @sql;
```

SQLQuery8.sql - A...ESKTOP\Ayush (66))*

```sql
-- Script to extract logins from PROD

DECLARE @sql NVARCHAR(MAX) = '';



SELECT @sql += 'CREATE LOGIN [' + l.name + ']

WITH PASSWORD = ' + CONVERT(VARCHAR(MAX), l.password_hash, 1) + ' HASHED, SID = ' +
CONVERT(VARCHAR(MAX), l.sid, 1) + ',

DEFAULT_DATABASE = [' + l.default_database_name + '];' + CHAR(13)

FROM sys.sql_logins l

WHERE l.is_disabled = 0;



PRINT @sql;
```

- ❑ Copy the Output of this Query and Run this on DEV

## ❖ Important Note

If some of these users (logins) already exist in the development environment, running the scripts as-is will result in an error, because SQL Server will not allow the creation of duplicate logins with the same name or SID.

To handle this situation, you can modify the script to check whether a login already exists before attempting to create it. Here's how you can adjust your script:

### Modify the Script to Prevent Duplicates

You can wrap the `CREATE LOGIN` statement with an `IF NOT EXISTS` clause to check whether the login already exists.

**Updated Script:**

```
If not exists(Select 1 from sys.server_principals where name = 'sa')

Begin

CREATE LOGIN [sa]

WITH PASSWORD =
0x02002AB1DE0CE84CCD5AC794ABF229E60195388CBA809D74B124A07A
F17132EE50DAD5E1D06AE8625FCFC6DD75F59F841789A1538930108A2226
340D6EBDD4AF83C7910DED19 HASHED, SID = 0x01,

DEFAULT_DATABASE = [master];

End

If not exists(Select 1 from sys.server_principals where name = 'Ayush')

Begin

CREATE LOGIN [Ayush]

WITH PASSWORD =
0x02000F2B67D155D39C0C846A109487DAF9A2F7339F66C917C58F4F8DE48
4821A048C80FD3F2E3A3B8369809AF27E5C0F4D35F9ADFE51A34C8D3C50
38EFD4ACF5E49E229B7B00 HASHED, SID =
0xC22117F72F5CA740BD175E85E2A4AB60,

DEFAULT_DATABASE = [master];

End

If not exists(Select 1 from sys.server_principals where name = 'Ayush2')

Begin

CREATE LOGIN [Ayush2]

WITH PASSWORD =
0x020053F0F073880CE2244E885BBEFCE7E2AFD59E628515173DA77B495D
```

AC9FB8C5CEE61A347CB4850D3280E63D95213F15DB00545EF5CC4E960C9F4735E6089DB2EFDCC16ACB HASHED, SID = 0xCBC2F25DBAF73340AE840141D457160B,

DEFAULT_DATABASE = [master];

End

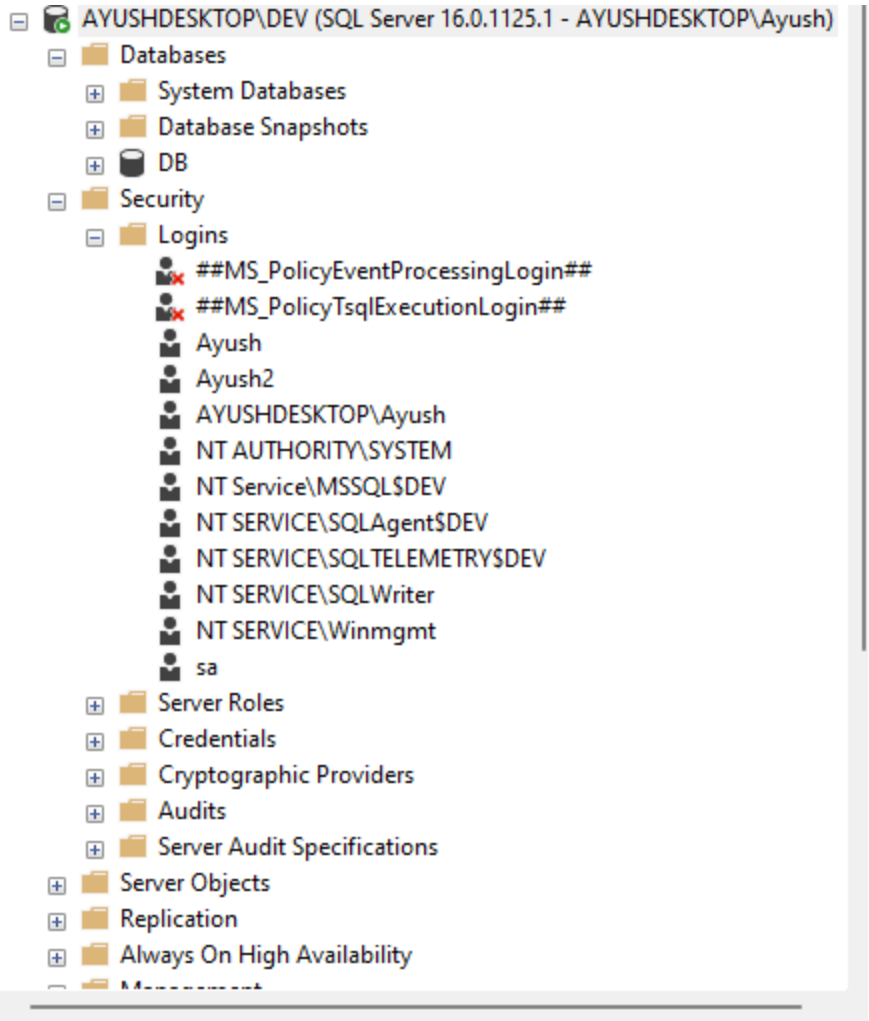
### Additional Step: Fixing Orphaned Users

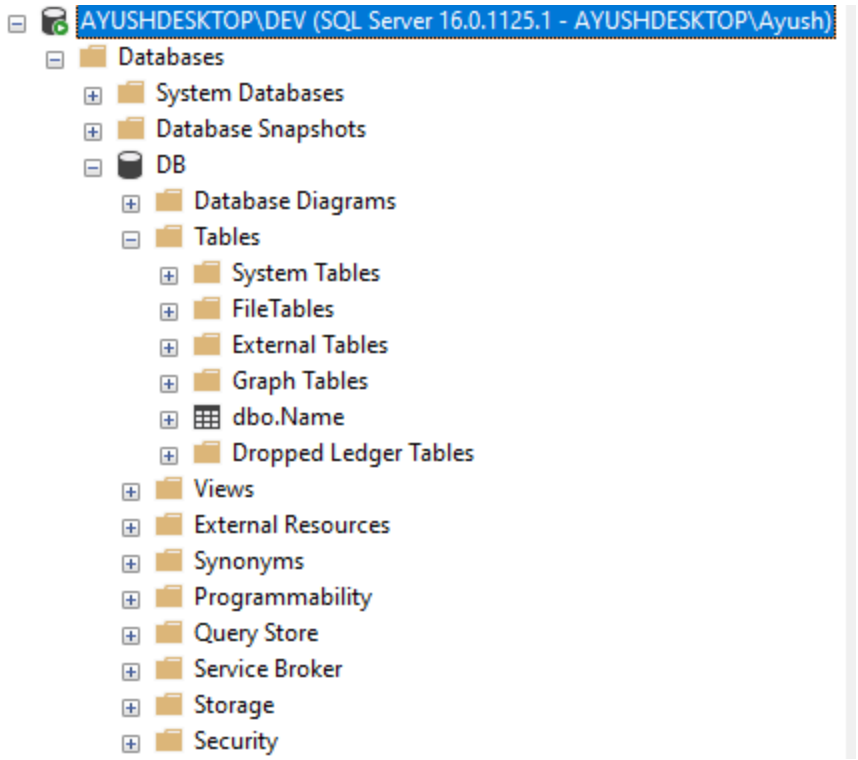
Even after successfully importing the logins, you may encounter orphaned users in the databases. If this happens, use the following command to remap the users to the correct login:

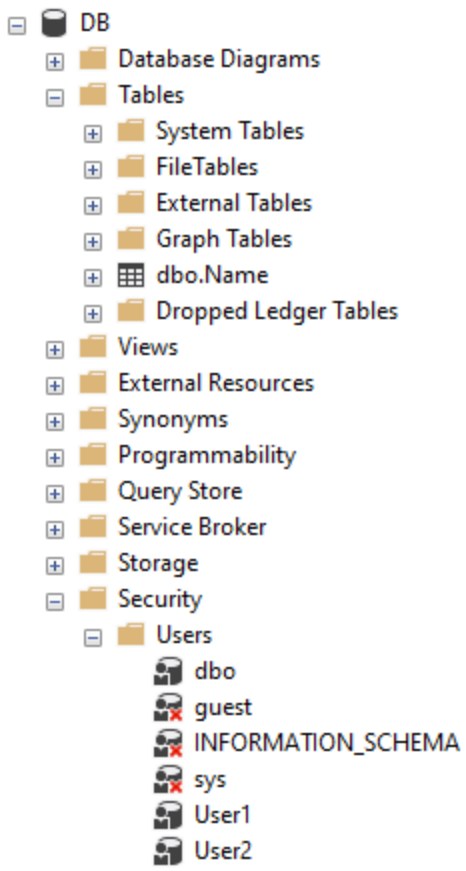
Use DB

EXEC sp_change_users_login 'Auto_fix','Username'


- ❑ Database Restore Completed


AYUSHDESKTOP\DEV (SQL Server 16.0.1125.1 - AYUSHDESKTOP\Ayush)
Databases
System Databases
Database Snapshots
DB
Security
Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
Ayush
Ayush2
AYUSHDESKTOP\Ayush
NT AUTHORITY\SYSTEM
NT Service\MSSQL$DEV
NT SERVICE\SQLAgent$DEV
NT SERVICE\SQLTELEMETRY$DEV
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa
Server Roles
Credentials
Cryptographic Providers
Audits
Server Audit Specifications
Server Objects
Replication
Always On High Availability


AYUSHDESKTOP\DEV (SQL Server 16.0.1125.1 - AYUSHDESKTOP\Ayush)
Databases
System Databases
Database Snapshots
DB
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Name
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security




## Lessons Learned

This experience reinforced the importance of meticulous planning and proactive problem-solving in database administration. Key takeaways include:

- Always verify environmental configurations and permissions before initiating database operations.
- Maintain a checklist of steps to address common errors swiftly.

## Conclusion

Database refreshes are crucial for maintaining the integrity and functionality of development environments. By sharing this detailed guide, I hope to assist fellow SQL Server professionals in enhancing their operations and troubleshooting skills.

## Call to Action

I encourage professionals in my network to share their experiences and tips on managing database refreshes. Let’s engage in a discussion to broaden our understanding and improve our practices.