ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo Lab 5

Mã hóa dữ liệu TDE

Môn học: Bảo mật cơ sở dữ liệu

CSC15002_22MMT

Sinh viên:
Nguyễn Hồ Đăng Duy
Phạm Quang Duy

Giảng viên hướng dẫn:
Nguyễn Đình Thúc
Nguyễn Thị Hường
Lê Trọng Anh Tú

Mục lục

1 Thông tin

1.1 Thông tin sinh viên

Nhóm gồm có 2 thành viên:

- 22127085 - Nguyễn Hồ Đăng Duy - 22127085@student.hcmus.edu.vn
- 22127088 - Phạm Quang Duy - 22127088@student.hcmus.edu.vn

1.2 Phân công nhiệm vụ

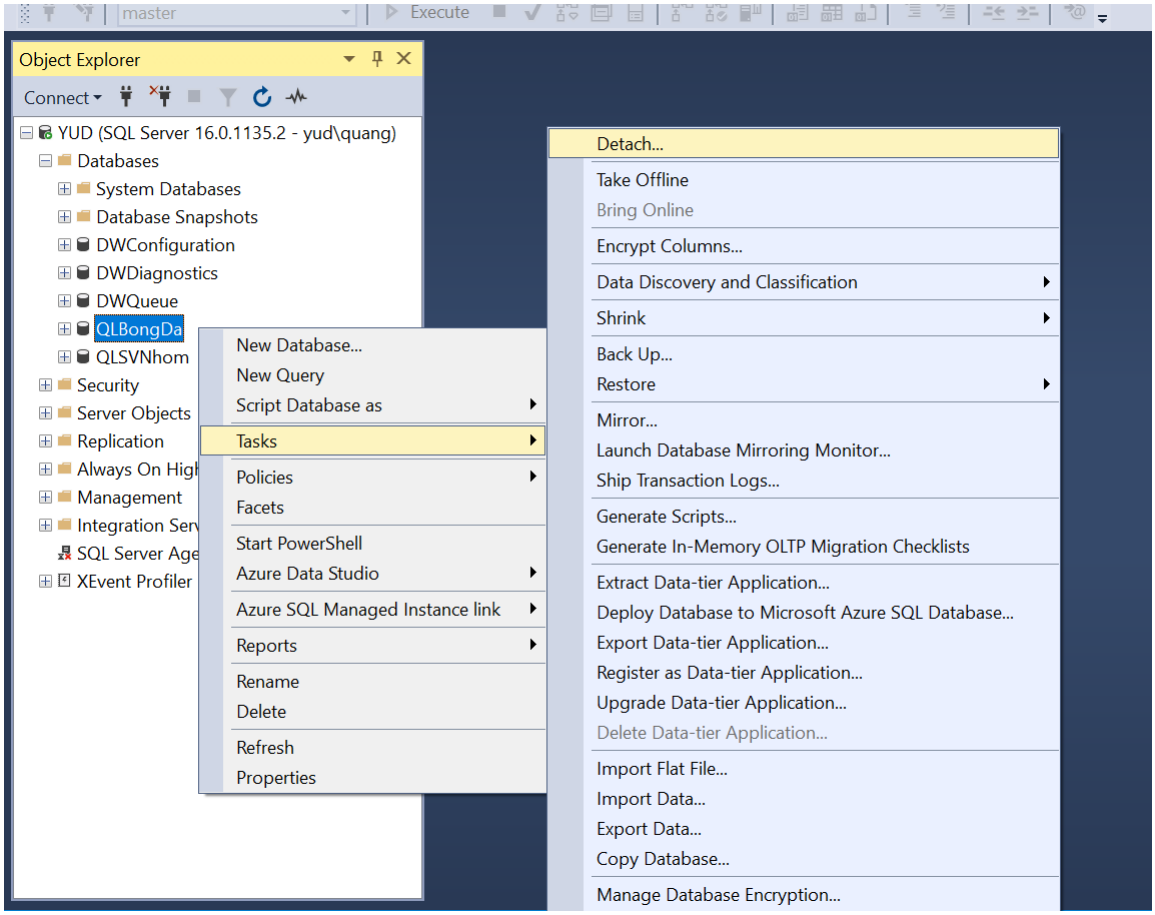
Sinh viên	Các câu đã làm	Tiến độ
Nguyễn Hồ Đăng Duy	d, e, f, g	100%
Phạm Quang Duy	a, b, c, report	100%

Bảng 1: Bảng phân công nhiệm vụ

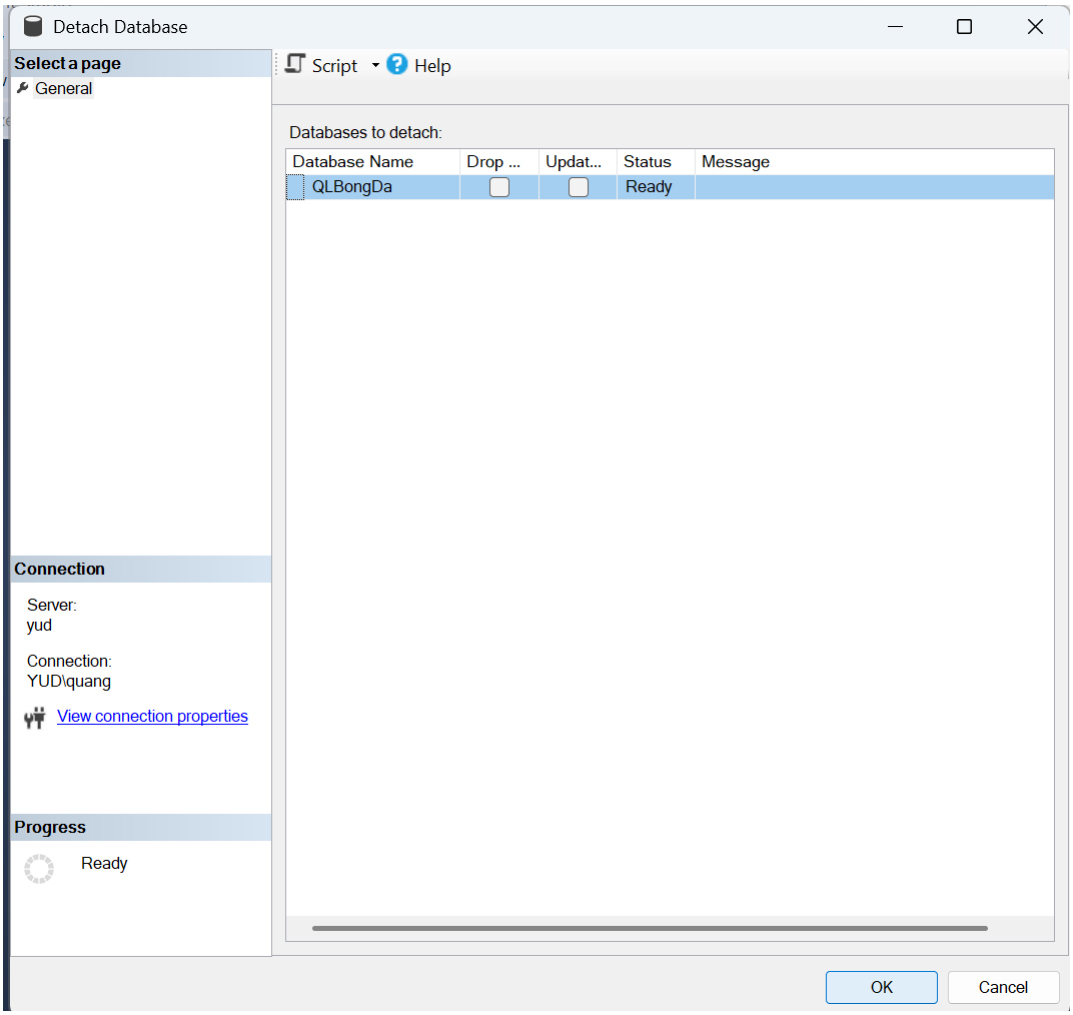
2 Detach - Attach không sử dụng TDE

2.1 Detach

- Mở SSMS và kết nối đến SQL Server instance A (Server "YUD").
- Mở thư mục **Databases** trong Object Explorer.
- Click phải vào database cần detach (QLBongDa).
- Chọn **Tasks → Detach...**.



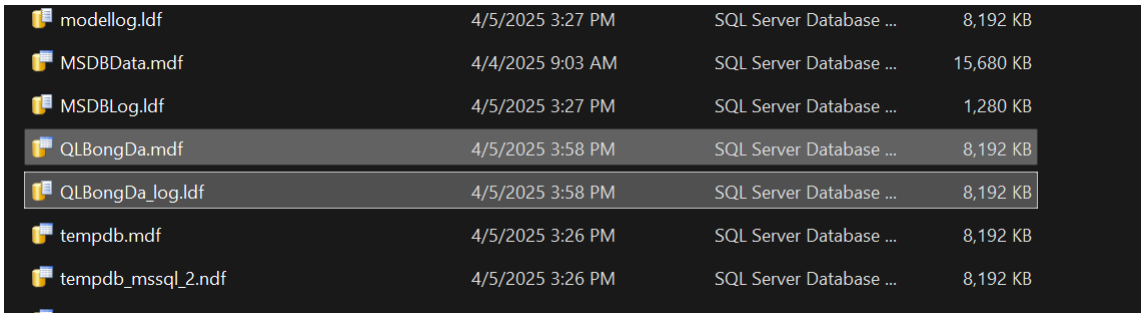
- Trong cửa sổ hiện ra:



- Nhấn **OK** để hoàn tất quá trình detach.

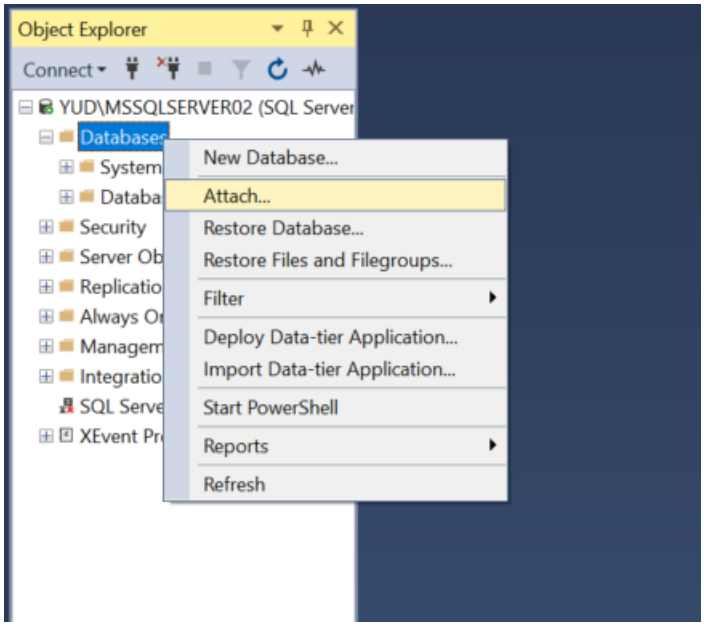
2.2 Attach

- Chuyển 2 file database: **".mdf"** và **".ldf"** từ folder DATA của server A (YUD) vào folder DATA của server B (YUD\MSSQLSERVER02)

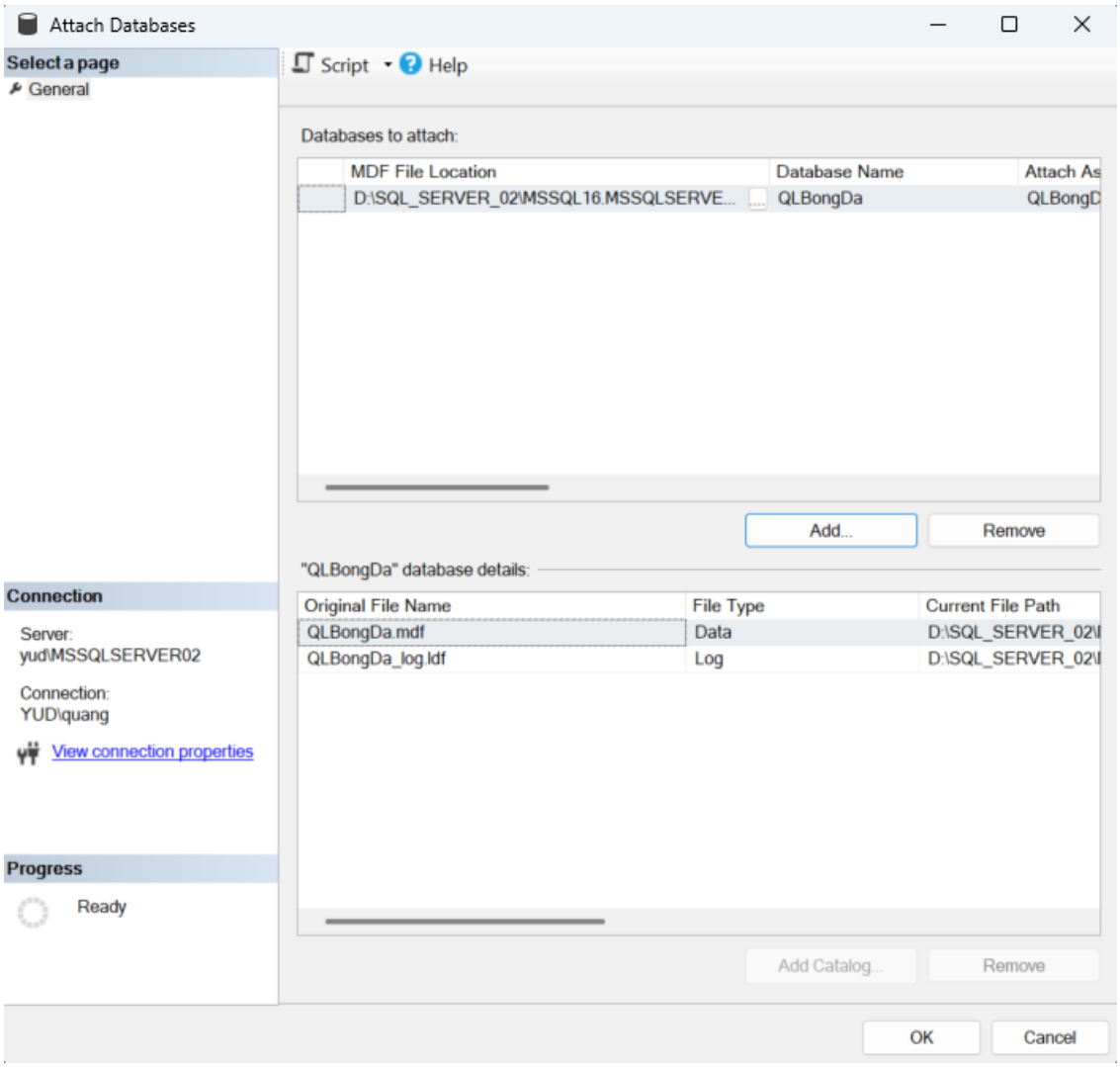


- Mở SSMS và kết nối đến SQL Server instance B (Server "YUD\MSSQLSERVER02").
- Mở thư mục **Databases** trong Object Explorer.

- Click phải vào mục **Databases** → **Attach...**.



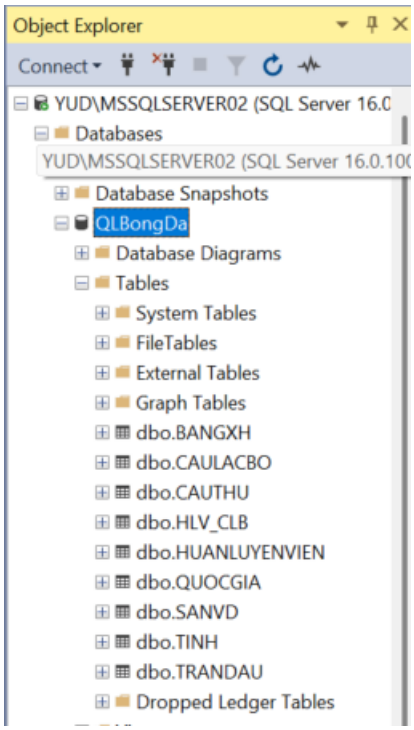
- Chọn **Add** và tìm đến file `.mdf`.
- SQL Server sẽ tự nhận diện file `.ldf` nếu có cùng đường dẫn.

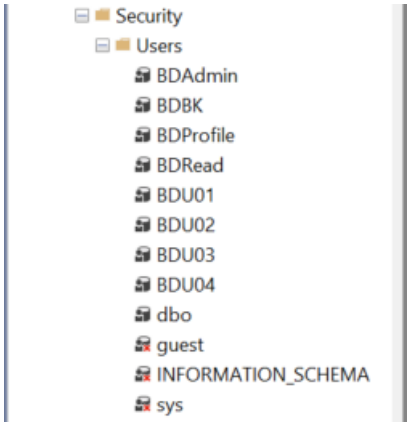


- Nhấn **OK** để attach.

2.3 Nhận xét

- Detach/Attach DB không TDE vẫn giữ nguyên toàn bộ schema và dữ liệu bên trong như bảng, stored procedures, function, views, users...



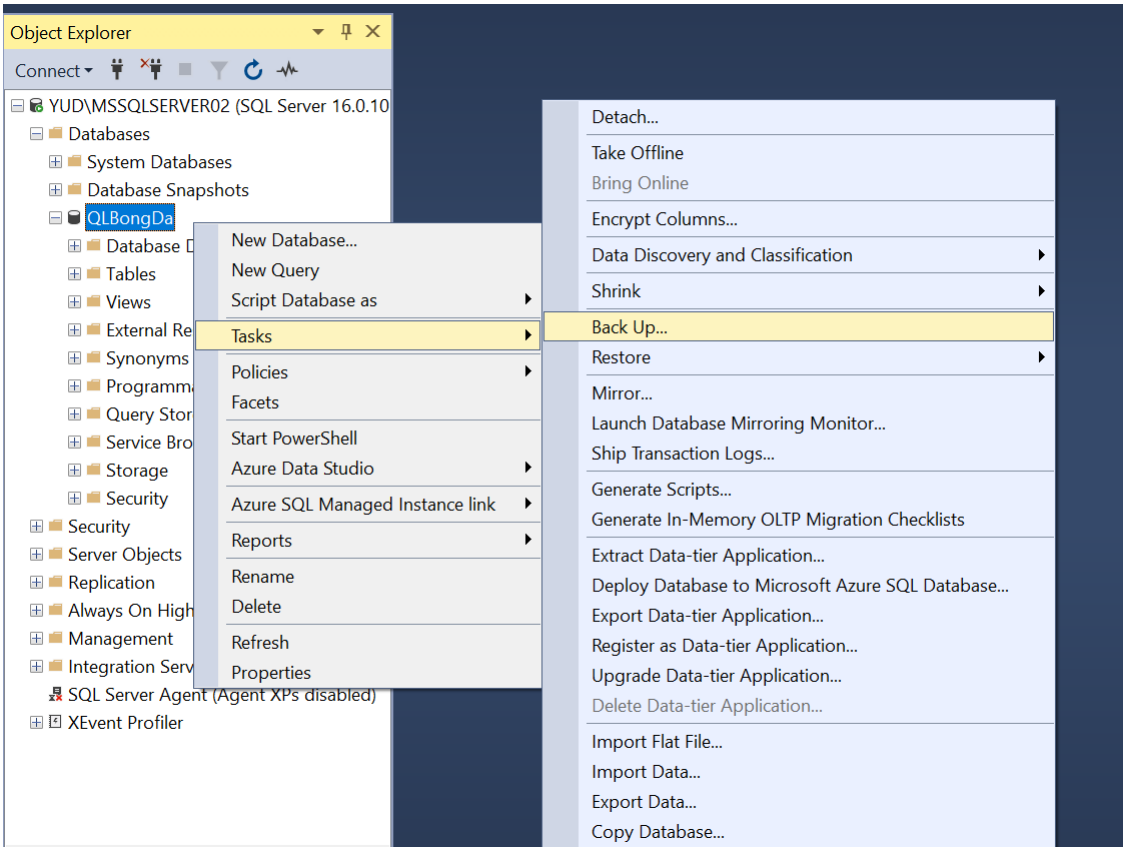


- Tuy nhiên, login ở cấp server không được giữ, cần khôi phục hoặc tạo lại trên server mới do có thể Login tương ứng trên Server B không tồn tại → gây lỗi khi user login.

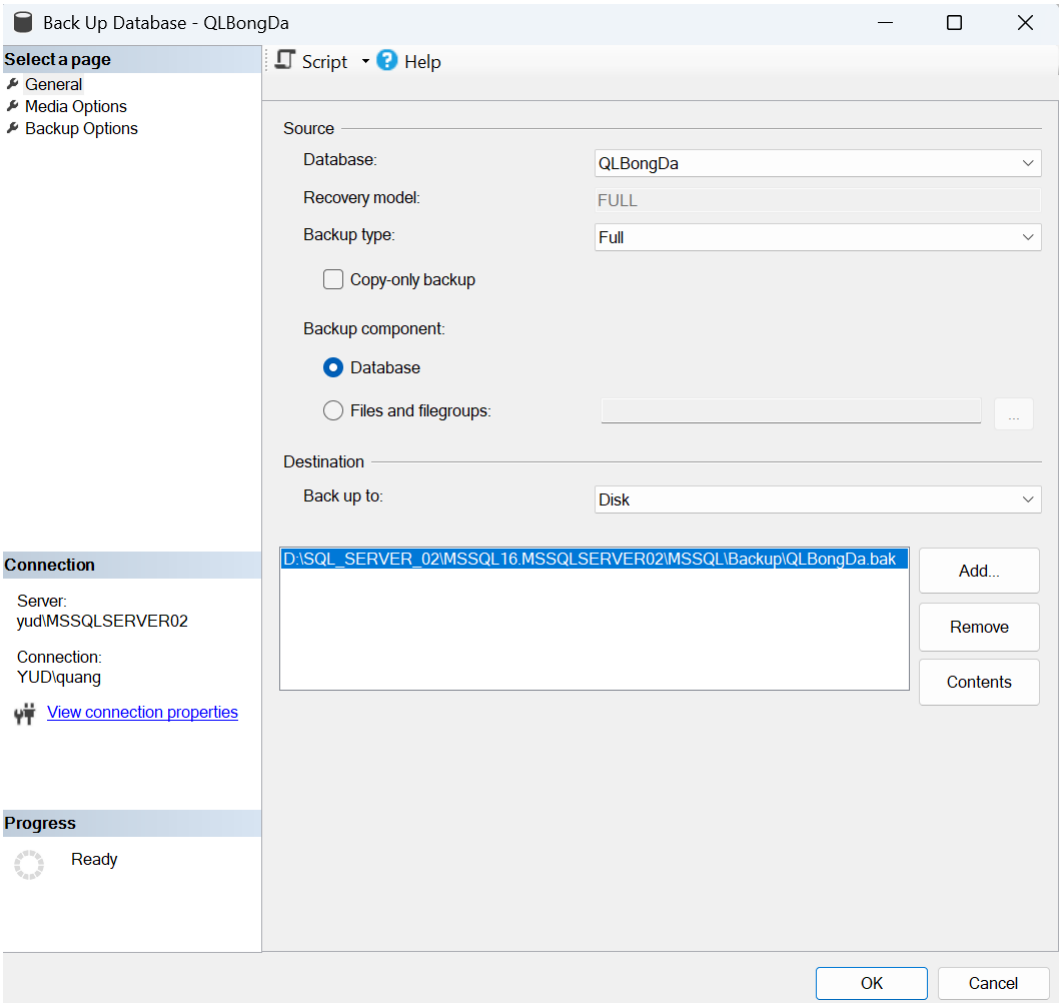
3 Backup - Restore không sử dụng TDE

3.1 Backup

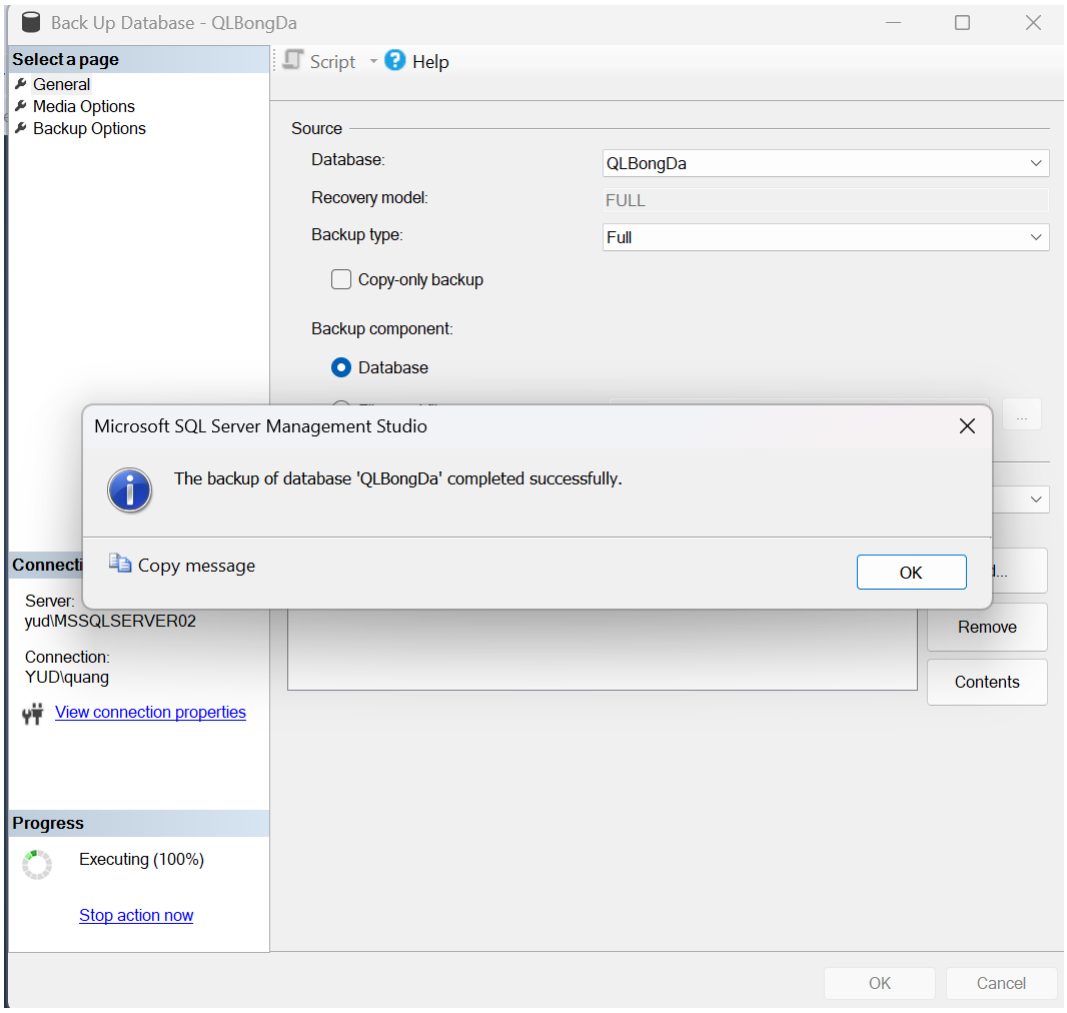
- Mở SSMS và kết nối đến SQL Server A ("YUD\MSSQLSERVER02").
- Mở **Object Explorer** và click phải vào database cần backup.
- Chọn **Tasks → Back Up...**.



- Trong cửa sổ Backup:
  - Chọn **Backup type**: Full / Differential / Transaction Log.
  - Kiểm tra đường dẫn lưu file backup (.bak).

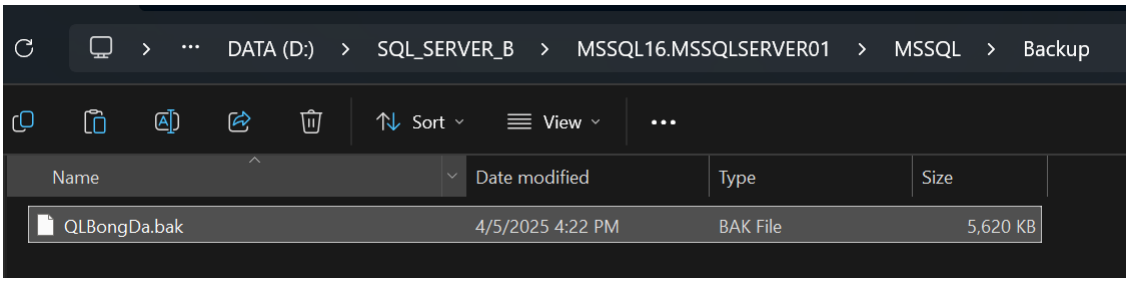


- Nhấn **OK** để thực hiện backup.

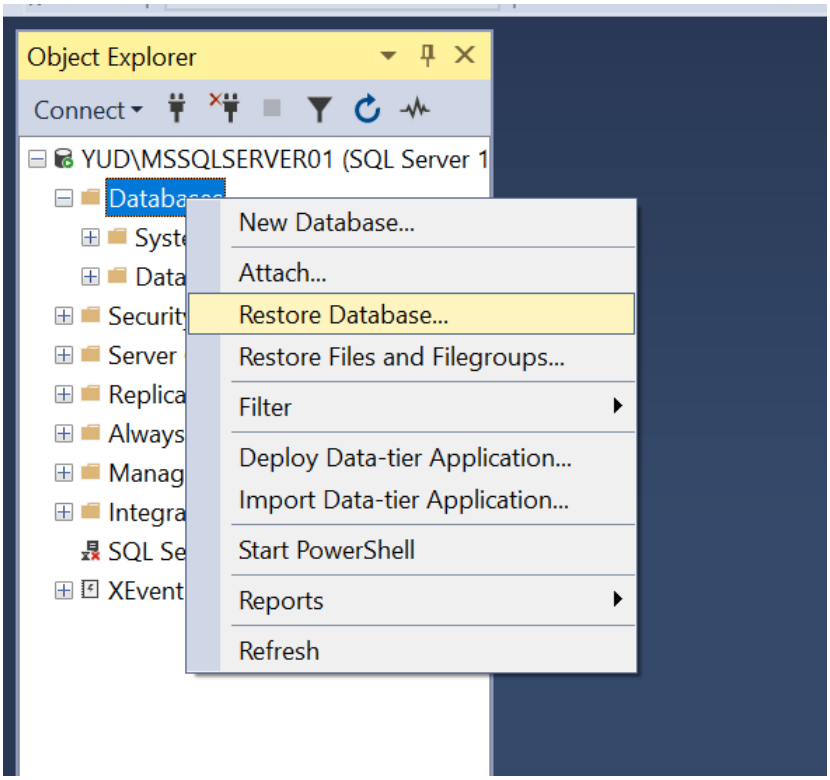


3.2 Restore

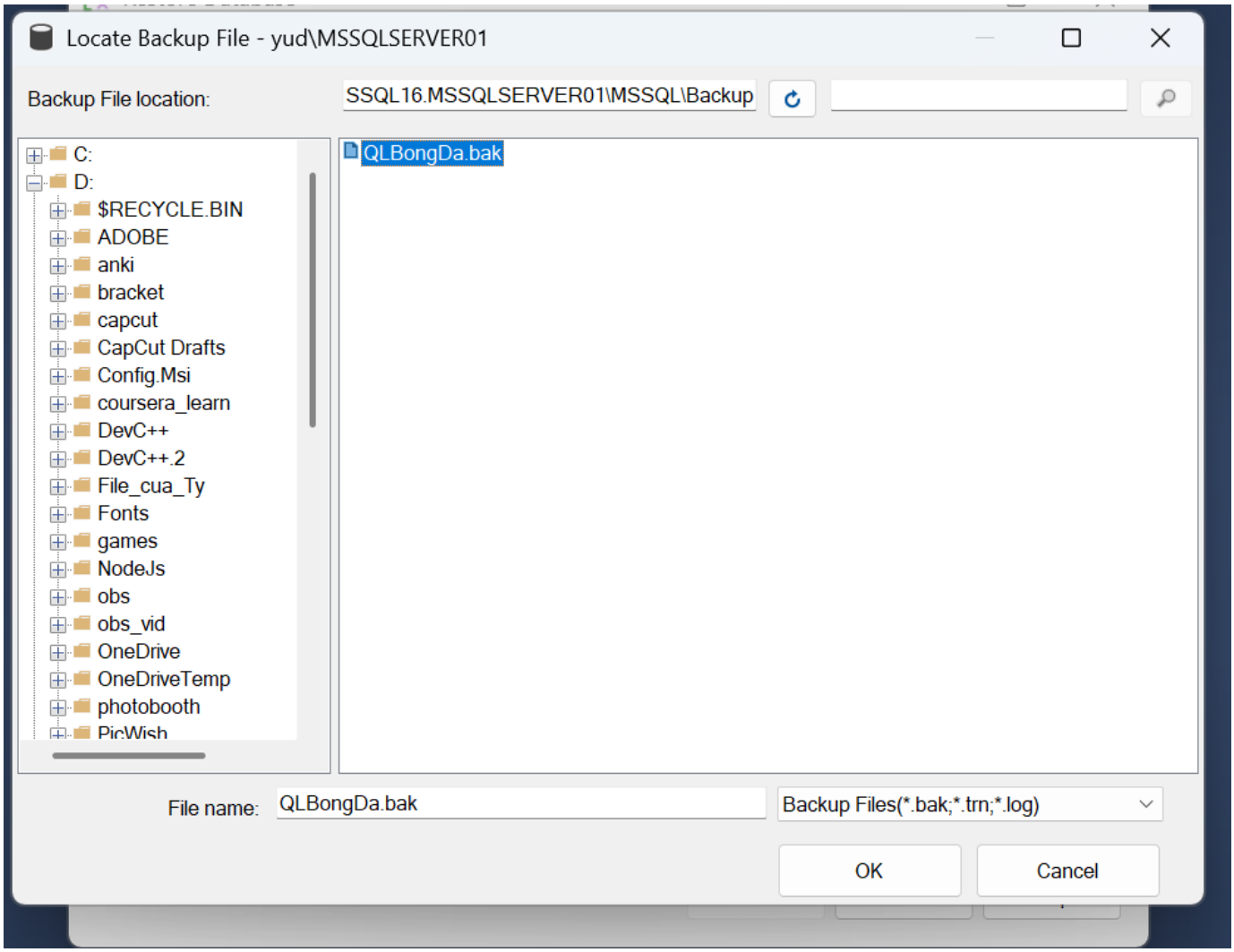
- Chuyển file **".bak"** từ folder BACKUP của server A (YUD\MSSQLSERVER02) vào folder BACKUP của server B (YUD\MSSQLSERVER01)

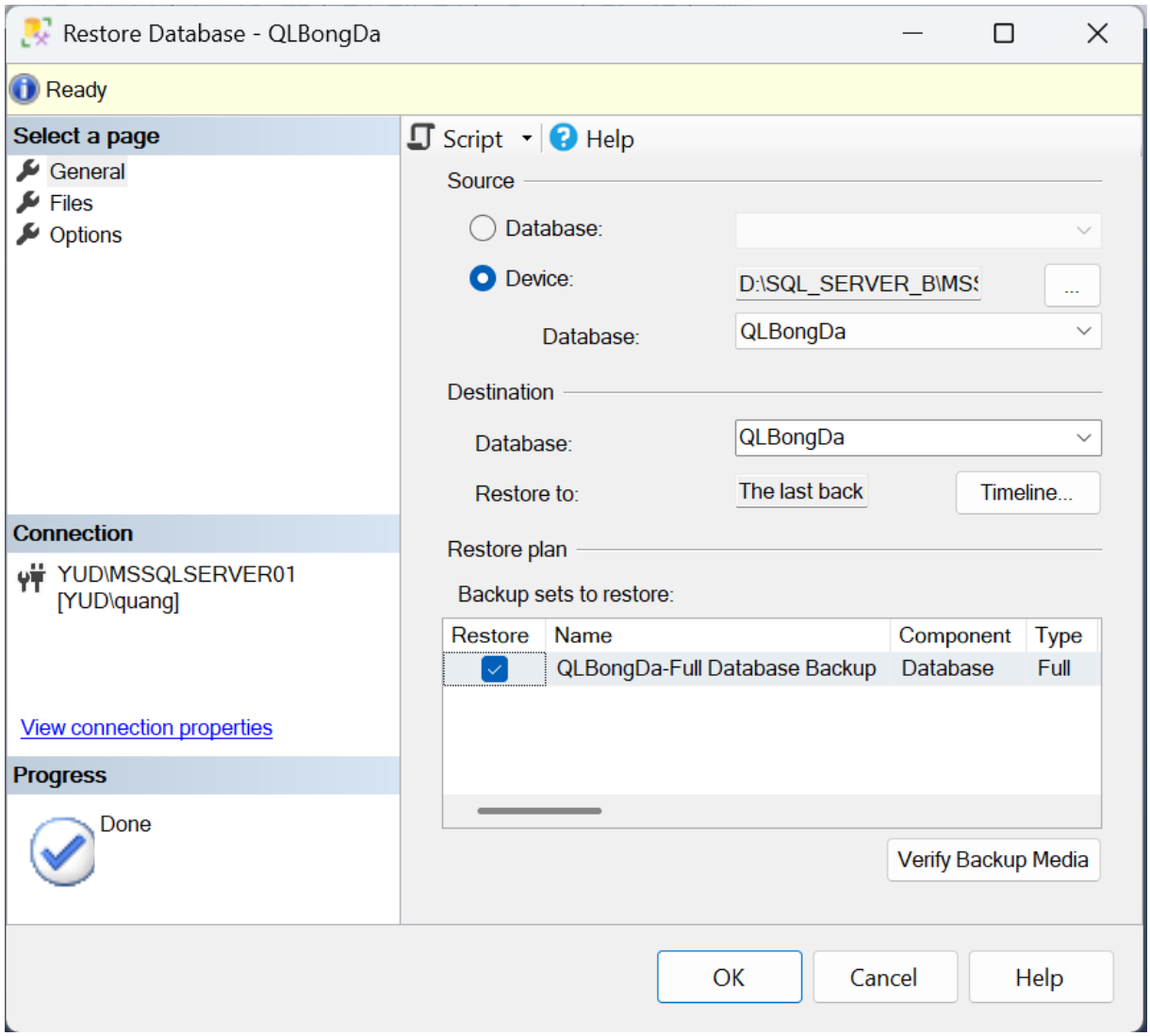


- Mở SSMS và kết nối đến SQL Server B ("YUD\MSSQLSERVER01").
- Click phải vào mục **Databases → Restore Database...**.

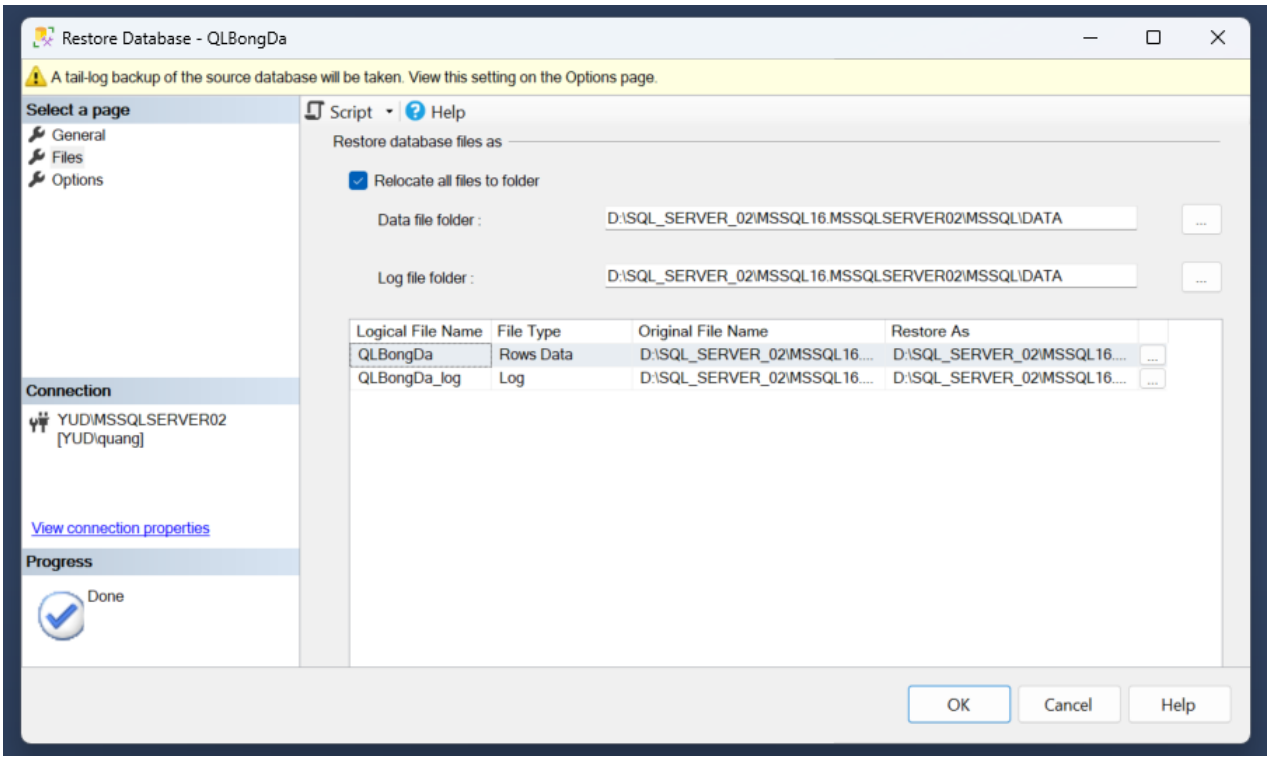


- Chọn **Device** → **Add** → chọn file `.bak`.


Locate Backup File - yud\MSSQLSERVER01
Backup File location:
SSQL16.MSSQLSERVER01\MSSQL\Backup
C:
D:
$RECYCLE.BIN
ADOBE
anki
bracket
capcut
CapCut Drafts
Config.Msi
coursera_learn
DevC++
DevC++.2
File_cua_Ty
Fonts
games
NodeJs
obs
obs_vid
OneDrive
OneDriveTemp
photobooth
PicWish
QLBongDa.bak
File name:
QLBongDa.bak
Backup Files(*.bak;*.trn;*.log)
OK
Cancel

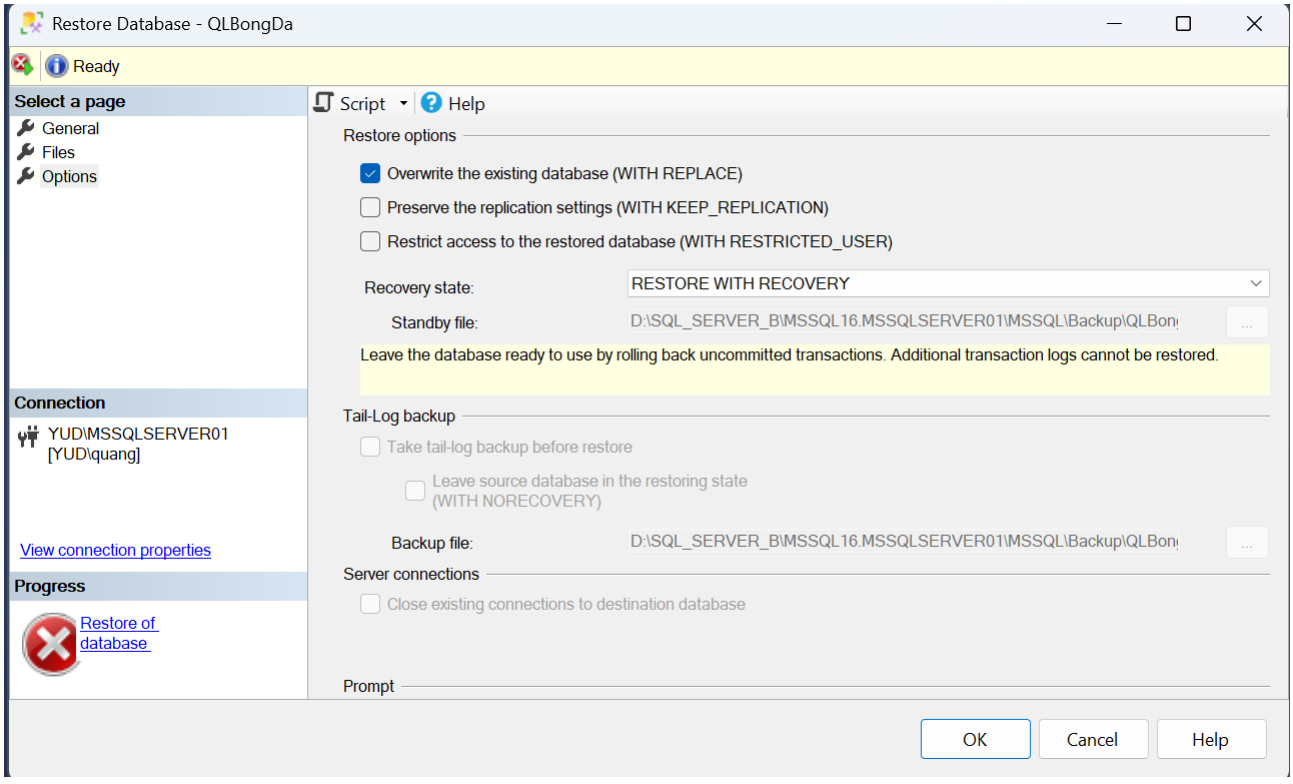


- Tùy chọn:
  - Trong thẻ "File": Chọn "Reallocate all files to folder"

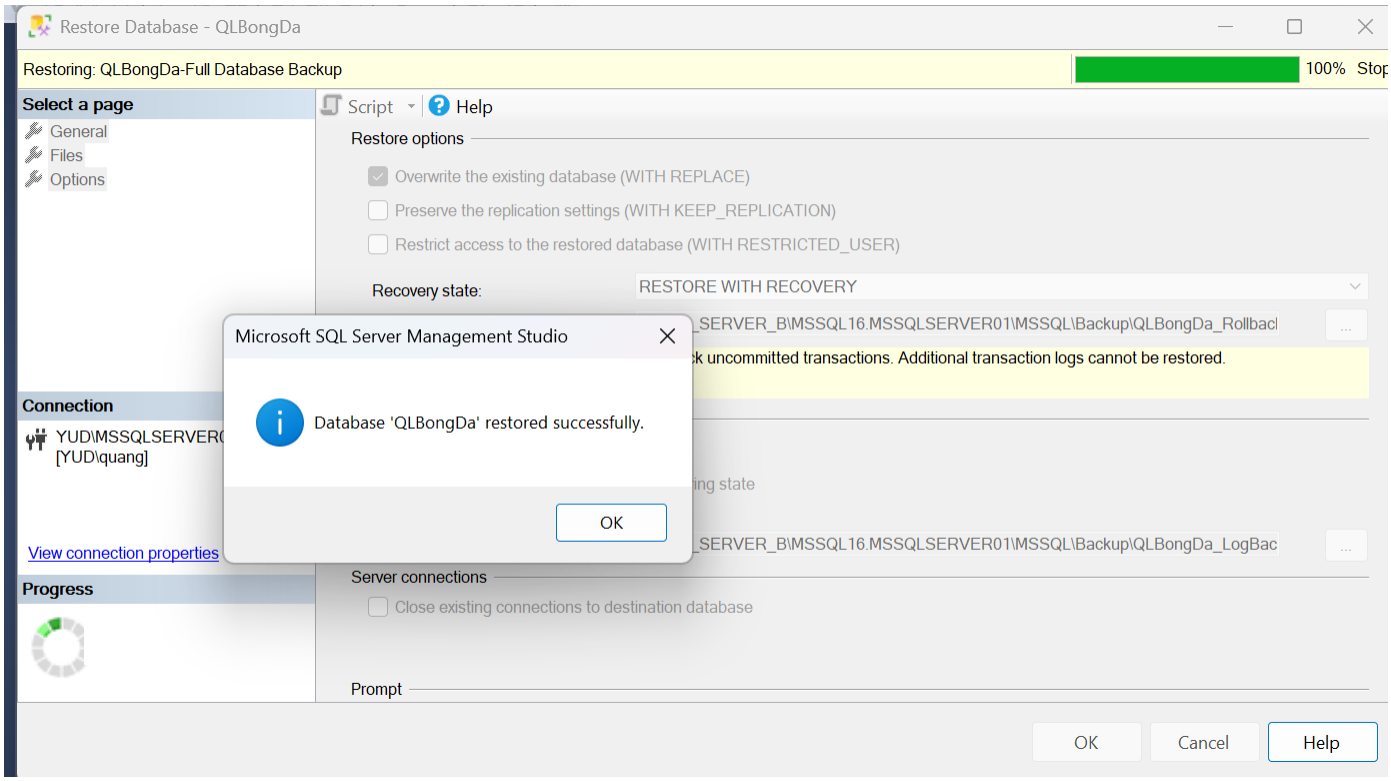


– Trong thẻ "Options": Restore with **RECOVERY**: phục hồi hoàn toàn (mặc định).

– Restore with **NORECOVERY**: dùng trong khôi phục theo chuỗi (ví dụ kèm file log) và chọn "Overwrite the existing databse".



- Nhấn **OK** để thực hiện restore.



3.3 Nhận xét

Chi tiết các đối tượng giữ nguyên sau khi Restore:

- Tables, Views, Stored Procedures, Triggers → Giữ nguyên đầy đủ như trên Server A
- Dữ liệu (Data) → Dữ liệu trong các bảng được khôi phục đầy đủ
- Users bên trong database (DB-level users) → Vẫn tồn tại trong database sau khi restore

Chi tiết các đối tượng không được giữ hoặc có thể lỗi sau khi Restore:

- SQL Server Logins (cấp server) → Không được lưu trong file .bak, nên sẽ không tồn tại trên Server B nếu chưa tạo thủ công
- Mapping giữa Login User (User mapping) → Có thể mất; dẫn đến lỗi "orphaned user" nếu login không được tạo lại đúng cách trên Server B

4 Viết script mã hóa TDE

```
-- Create Master Key in 'master' if doesnt exist
USE master;
GO
IF NOT EXISTS (
    SELECT * FROM sys.symmetric_keys WHERE symmetric_key_id = 101
)
BEGIN
    CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'QuangDuyDangDuy@2112708522127088';
    PRINT N'Master Key created.';
END
ELSE
    PRINT N'Master Key exists.';

-- Create certificate TDECert if doesnt exist
IF NOT EXISTS (
    SELECT * FROM sys.certificates WHERE name = 'TDECert'
)
BEGIN
    CREATE CERTIFICATE TDECert
    WITH SUBJECT = 'Certificate for TDE - QLBongDa';
    PRINT N'Certificate TDECert created.';
END
ELSE
    PRINT N'Certificate TDECert exists.';

-- Create Database Encryption Key and turn on TDE for QLBongDa
USE QLBongDa;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDECert;
GO

ALTER DATABASE QLBongDa SET ENCRYPTION ON;
GO
PRINT N'Database QLBongDa encrypted by TDE.';

-- Check encrypting status
SELECT
    db.name AS DatabaseName,
    db.is_encrypted AS IsEncrypted,
    dek.encryption_state AS EncryptionState,
    dek.percent_complete AS PercentComplete,
    dek.key_algorithm AS Algorithm,
    dek.key_length AS KeyLength
FROM sys.databases db
LEFT JOIN sys.dm_database_encryption_keys dek
    ON db.database_id = dek.database_id
WHERE db.name = 'QLBongDa';
```

4.1 Tạo Master Key trong database master nếu chưa có

```
USE master;
GO
IF NOT EXISTS (
    SELECT * FROM sys.symmetric_keys WHERE symmetric_key_id = 101
)
BEGIN
    CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'QuangDuyDangDuy@2112708522127088
    ';
    PRINT N'Master Key created.';
END
ELSE
    PRINT N'Master Key exists.';
```

Mục đích:

- Tạo Master Key – khóa gốc dùng để mã hóa các chứng chỉ và khóa khác.
- Bắt buộc phải có trước khi tạo Certificate.

4.2 Tạo Certificate để dùng cho TDE

```
IF NOT EXISTS (
    SELECT * FROM sys.certificates WHERE name = 'TDECert'
)
BEGIN
    CREATE CERTIFICATE TDECert
    WITH SUBJECT = 'Certificate for TDE - QLBongDa';
    PRINT N'Certificate TDECert created.';
END
ELSE
    PRINT N'Certificate TDECert exists.';
```

Mục đích:

- Tạo certificate để mã hóa Database Encryption Key (DEK) trong bước tiếp theo.

4.3 Tạo Database Encryption Key (DEK) và bật mã hóa TDE

```
USE QLBongDa;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDECert;
GO

ALTER DATABASE QLBongDa SET ENCRYPTION ON;
GO
PRINT N'Database QLBongDa encrypted by TDE.';
```

Mục đích:

- Tạo DEK để mã hóa dữ liệu trong database.

- Dùng thuật toán mã hóa mạnh: `AES_256`.
- Kích hoạt tính năng TDE cho database.

4.4 Kiểm tra trạng thái mã hóa của database

```sql
SELECT
    db.name AS DatabaseName,
    db.is_encrypted AS IsEncrypted,
    dek.encryption_state AS EncryptionState,
    dek.percent_complete AS PercentComplete,
    dek.key_algorithm AS Algorithm,
    dek.key_length AS KeyLength
FROM sys.databases db
LEFT JOIN sys.dm_database_encryption_keys dek
    ON db.database_id = dek.database_id
WHERE db.name = 'QLBongDa';
```

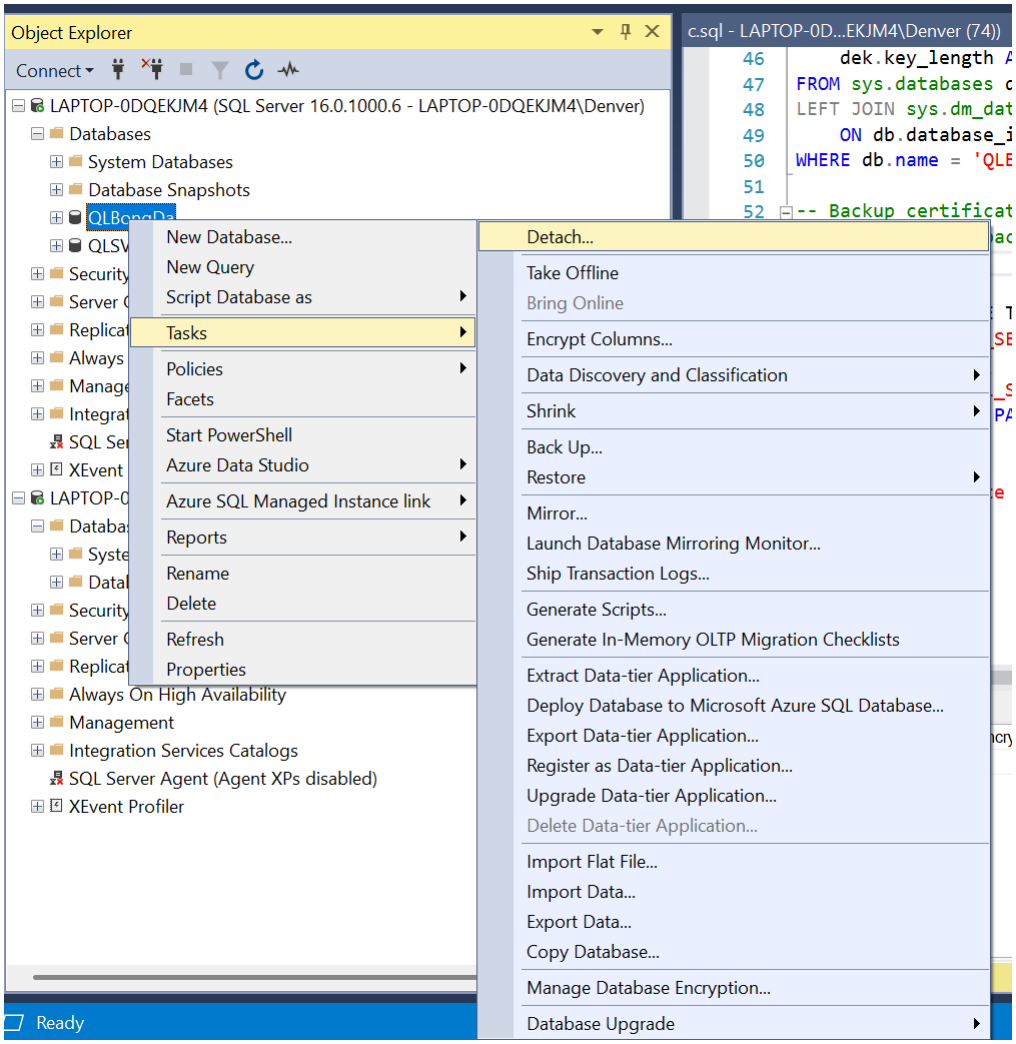
Mục đích:

- Kiểm tra trạng thái mã hóa của CSDL.
- `IsEncrypted = 1`: đã mã hóa.
- `encryption_state = 3`: quá trình mã hóa hoàn tất.

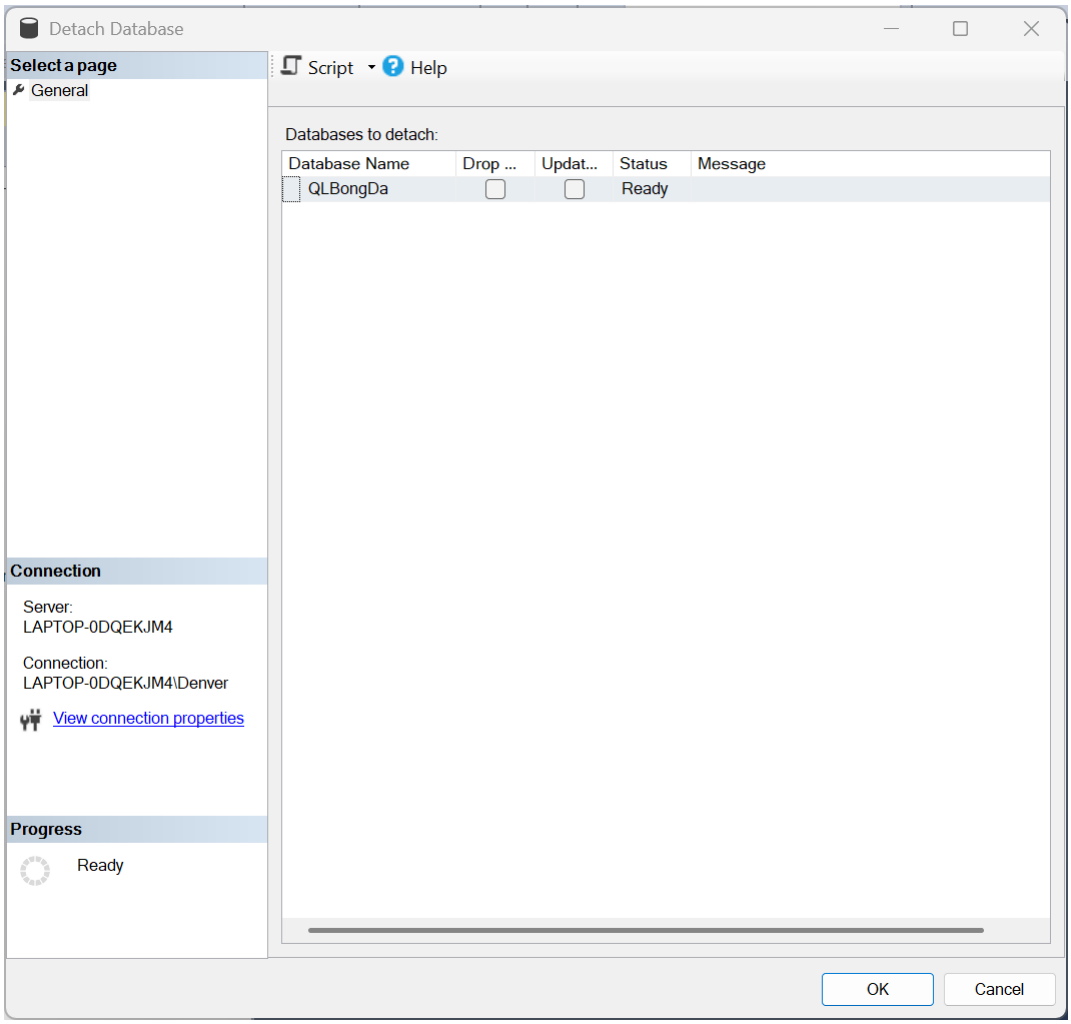
5 Detach - Attach sử dụng TDE

5.1 Detach

- Mở SSMS và kết nối đến SQL Server instance A (Server "LAPTOP-0DQEKJM4").
- Mở thư mục **Databases** trong Object Explorer.
- Click phải vào database cần detach (`QLBongDa`).
- Chọn **Tasks → Detach...**.



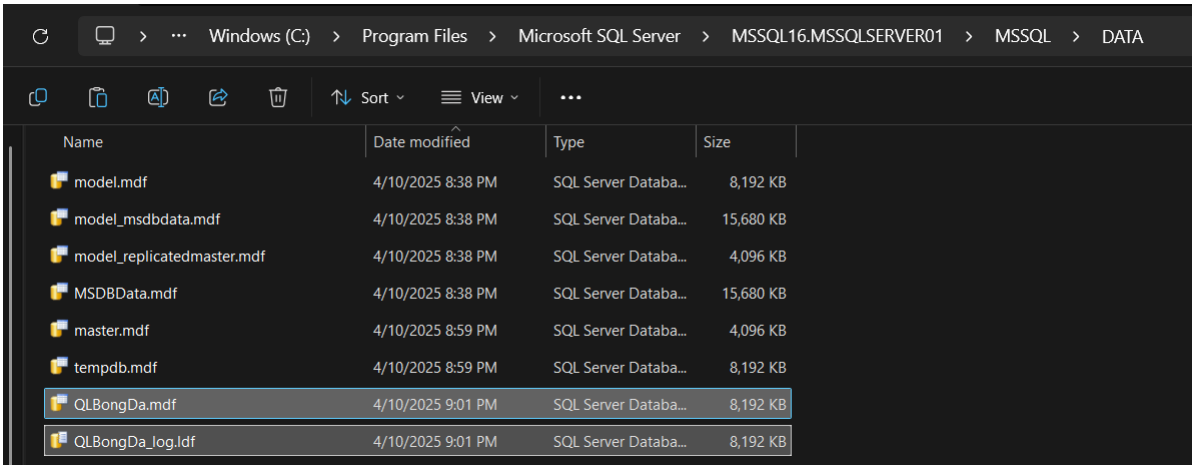
- Hiện ra:



- Nhấn **OK** để hoàn tất quá trình detach.

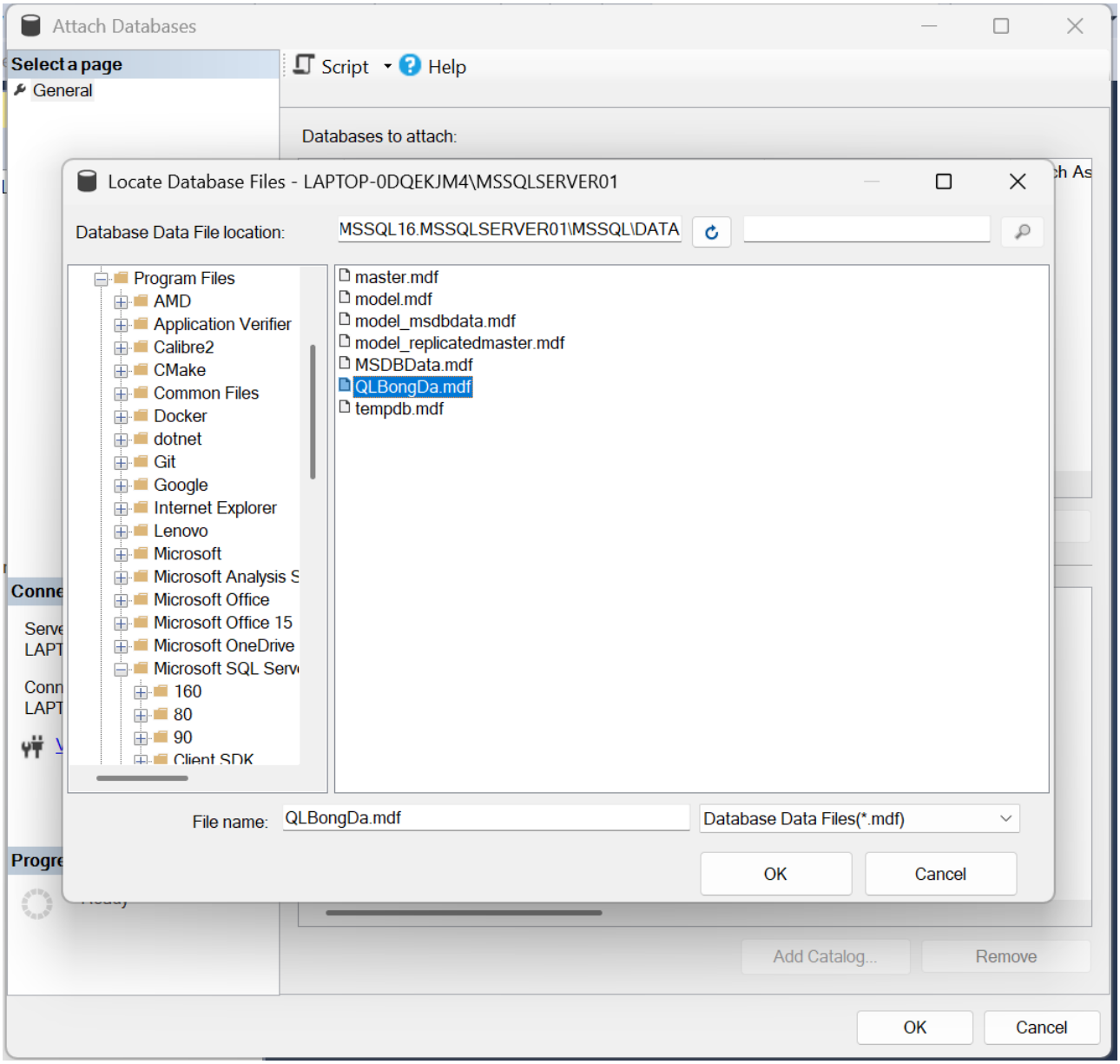
5.2 Copy file MDF và LDF

- Copy 2 file: `QLBongDa.mdf` và `QLBongDa_log.ldf` sang thư mục DATA của server B.

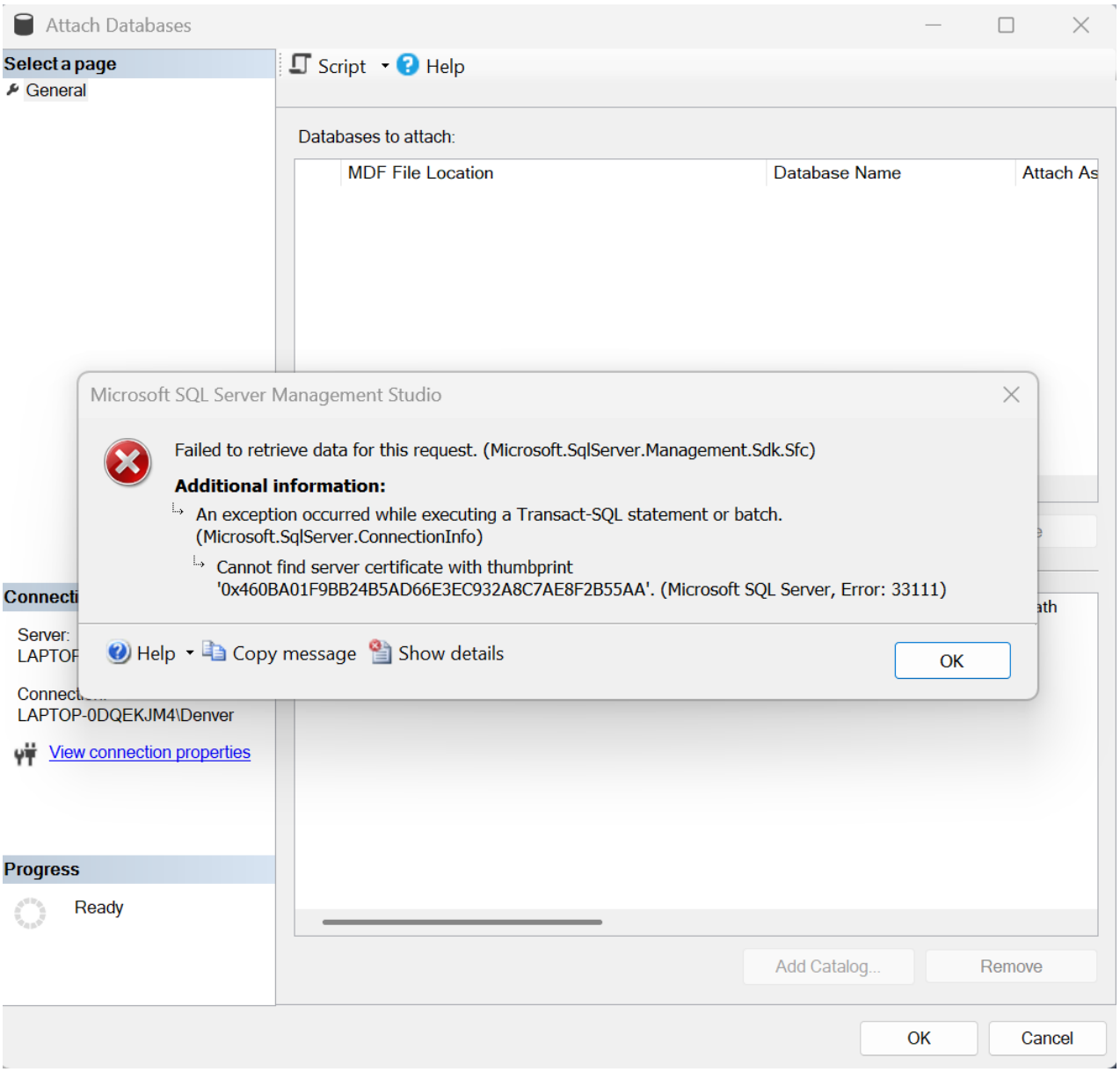


5.3 Attach

- Mở SSMS và kết nối đến SQL Server instance B (Server "LAPTOP-0DQEKJM4\MSSQLSERVER01").
- Mở thư mục **Databases** trong Object Explorer.
- Click phải vào mục **Databases** → **Attach...**.
- Chọn **Add** và tìm đến file `QLBongDa.mdf`.



- Nhấn **OK** để attach.
- Kết quả: xuất hiện lỗi không thể attach.



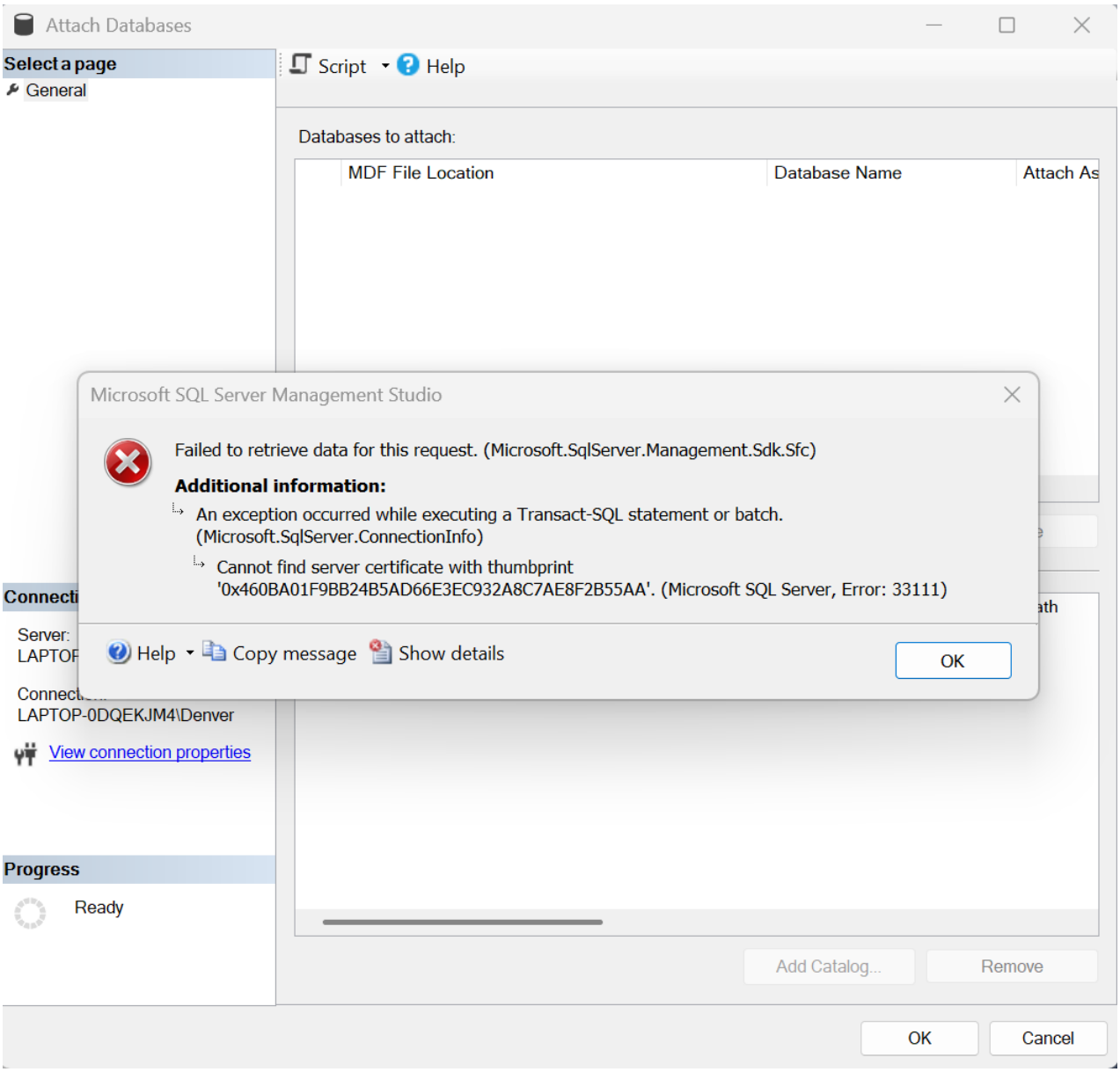
5.4 Nhận xét

- Khi Detach - Attach database có sử dụng TDE, nếu không thực hiện backup và restore certificate từ Server A sang Server B, thao tác attach sẽ **thất bại**.
- Lỗi hiển thị do SQL Server không tìm thấy certificate tương ứng để giải mã CSDL.

6 Xử lí lỗi Detach - Attach khi sử dụng TDE

6.1 Nguyên nhân lỗi

- Sau khi thực hiện **Detach** và copy file `.mdf`, `.ldf` từ Server A sang Server B, việc **Attach** database có sử dụng TDE trên Server B gặp lỗi.
- SQL Server thông báo không tìm thấy certificate dùng để mã hóa database:



Hình 1: Lỗi khi attach database TDE: thiếu certificate

- Đây là lỗi thường gặp khi di chuyển database đã mã hóa bằng TDE sang server khác mà không mang theo certificate tương ứng.

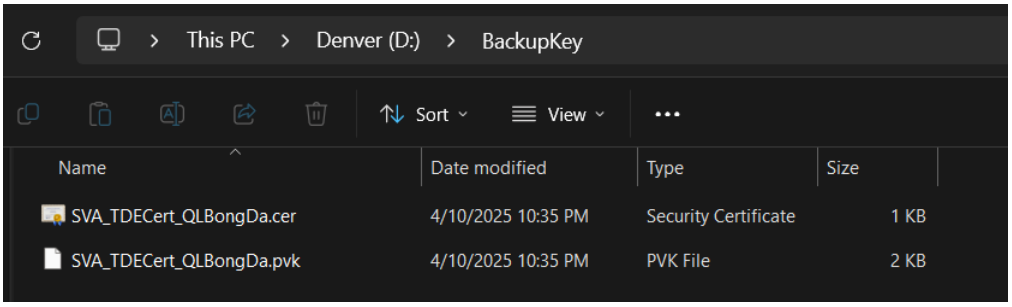
6.2 Giải thích

- Transparent Data Encryption (TDE) sử dụng certificate lưu trong `master` database để mã hóa **Database Encryption Key (DEK)**.
- Khi detach, file `.mdf` vẫn chứa dữ liệu đã mã hóa, nhưng certificate dùng để giải mã lại nằm ở Server A.
- Server B không có certificate tương ứng nên không thể mở được CSDL khi attach → gây lỗi.

6.3 Các bước xử lý lỗi

1. **Trên Server A:** Thực hiện **Backup Certificate** đang dùng cho TDE:
   - Tạo thư mục chứa file backup, ví dụ: `D:\BackupKey\`
   - Backup certificate ra file `.cer` và `.pvk` bằng đoạn code sau:

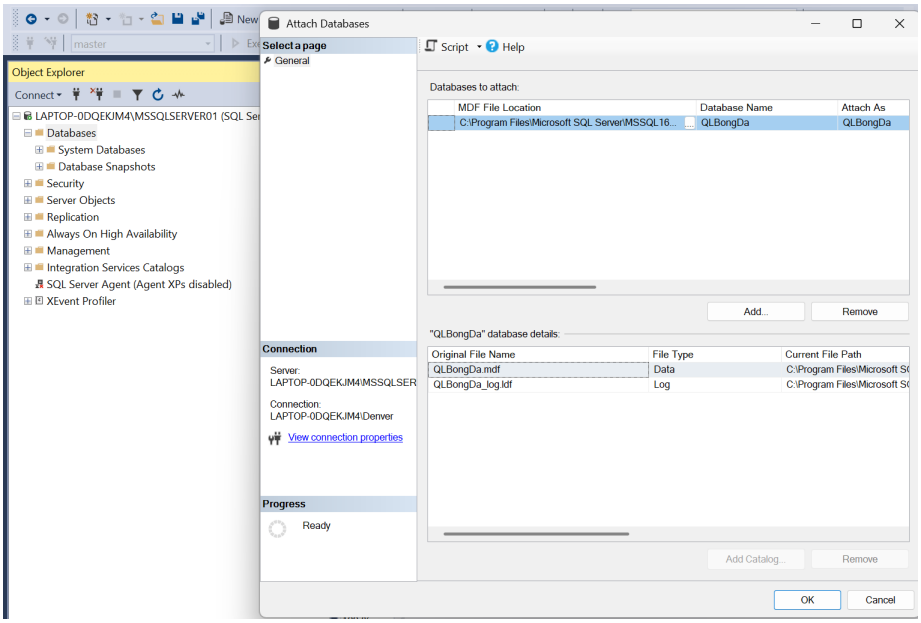
```
-- Backup certificate to use in other DB (Detach/Restore)
-- Create folder backup key for other server
USE master;
GO
BACKUP CERTIFICATE TDECert
TO FILE = 'D:\BackupKey\SVA_TDECert_QLBongDa.cer'
WITH PRIVATE KEY (
    FILE = 'D:\BackupKey\SVA_TDECert_QLBongDa.pvk',
    ENCRYPTION BY PASSWORD = '22127085_22127088'
);
GO
PRINT N'Certificate TDECert backup successfully.';
```

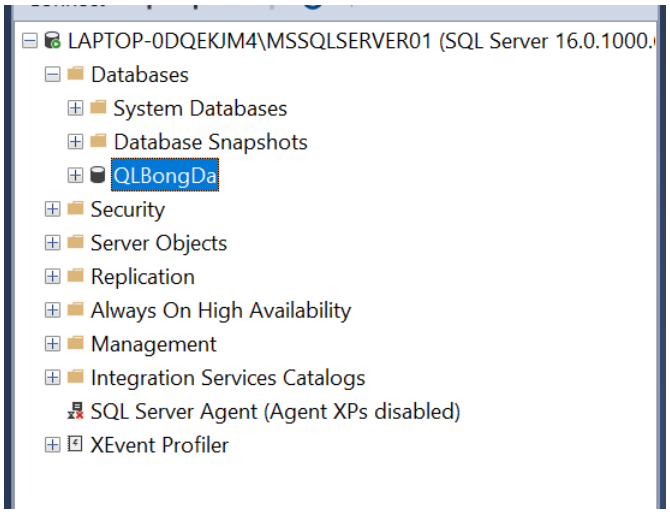


2. **Copy 2 file** `TDECert_QLBongDa.cer` và `TDECert_QLBongDa.pvk` sang Server B (ở đây là thư mục **NewBackupKey**)
3. **Trên Server B:** Khôi phục certificate từ các file đã sao chép:

```
USE master;
GO
CREATE CERTIFICATE TDEcert
FROM FILE = 'D:\NewBackup\SVA_TDECert_QLBongDa.cer'
WITH PRIVATE KEY (
  FILE = 'D:\NewBackup\SVA_TDECert_QLBongDa.pvk',
    DECRYPTION BY PASSWORD = '22127085_22127088');
GO
```

4. Thực hiện lại thao tác Attach database → Thành công.





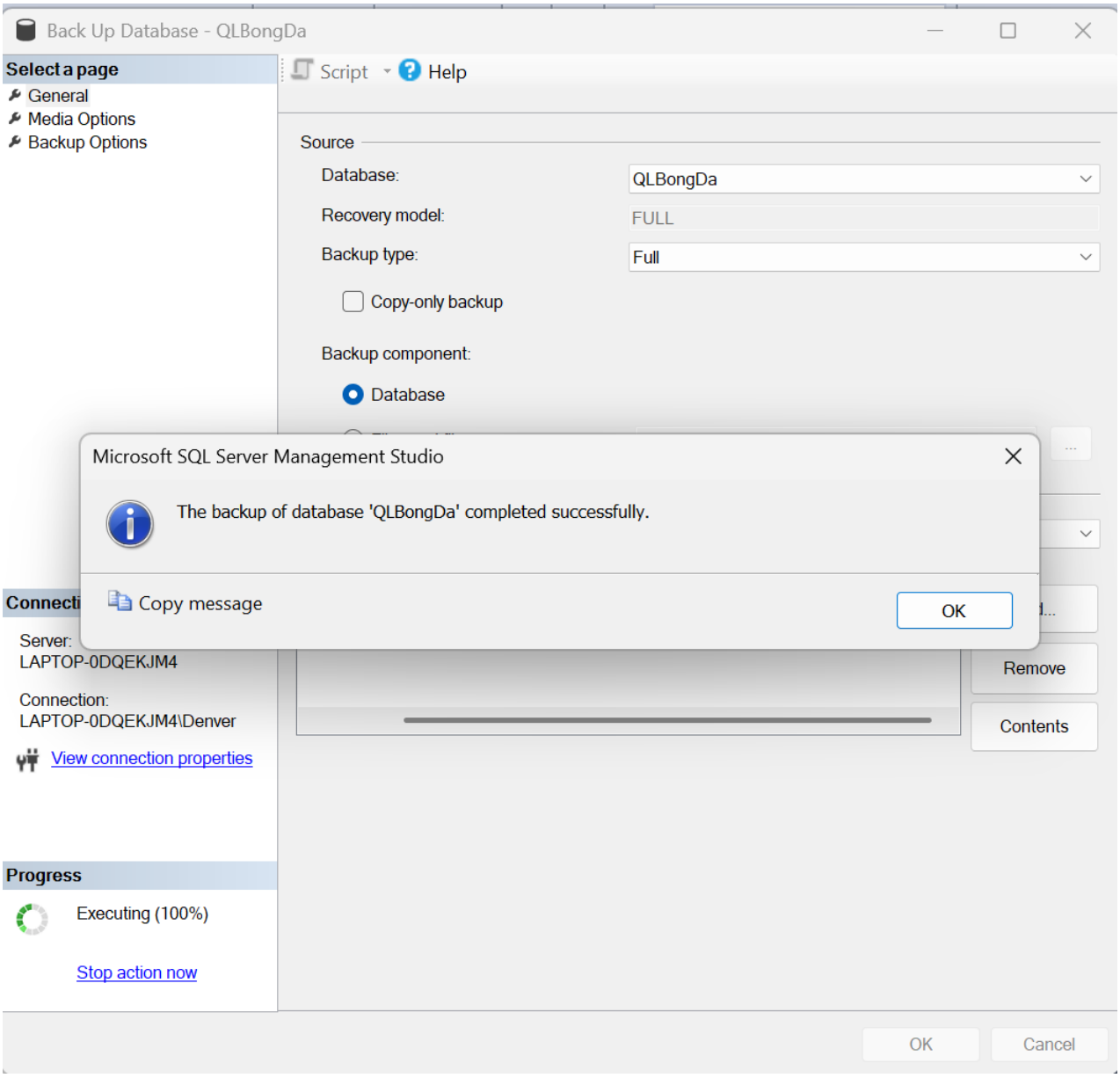
6.4 Kết luận

- Khi di chuyển database đã mã hóa bằng TDE, certificate gốc dùng để mã hóa/giải mã là yếu tố **bắt buộc** phải mang theo.
- Nếu không restore certificate tương ứng, mọi thao tác `Attach` hoặc `Restore` sẽ **thất bại**.
- Đây là cơ chế bảo mật giúp tránh việc đọc lén dữ liệu ngay cả khi file CSDL bị đánh cắp.
- Sau khi khôi phục certificate, việc attach diễn ra bình thường và toàn bộ dữ liệu được giữ nguyên.

7 Backup - Restore sử dụng TDE

7.1 Backup database có TDE

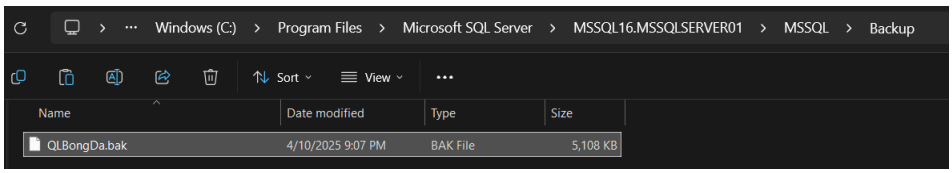
- Mở SSMS và kết nối đến SQL Server instance A (Server "LAPTOP-0DQEKJM4").
- Chuột phải vào database `QLBongDa` → chọn **Tasks** → **Back Up...**
- Chọn **Full Backup**, định vị file `.bak` sẽ lưu.
- Nhấn **OK** để thực hiện backup.



Hình 2: Backup database có sử dụng TDE

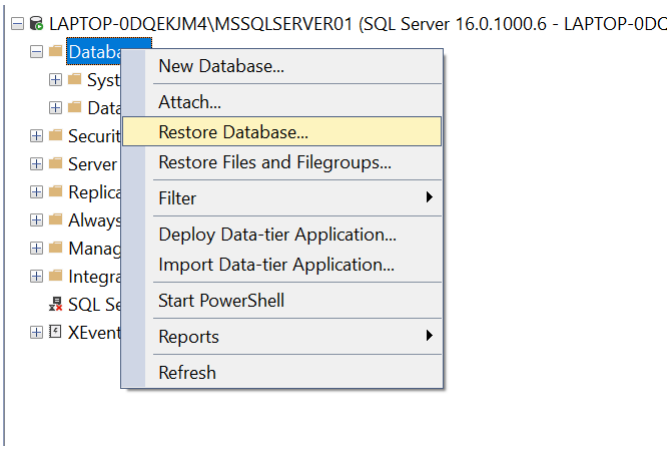
7.2 Copy file .bak và certificate sang Server B

- Copy file `QLBongDa.bak` từ Server A sang thư mục Backup trên Server B.

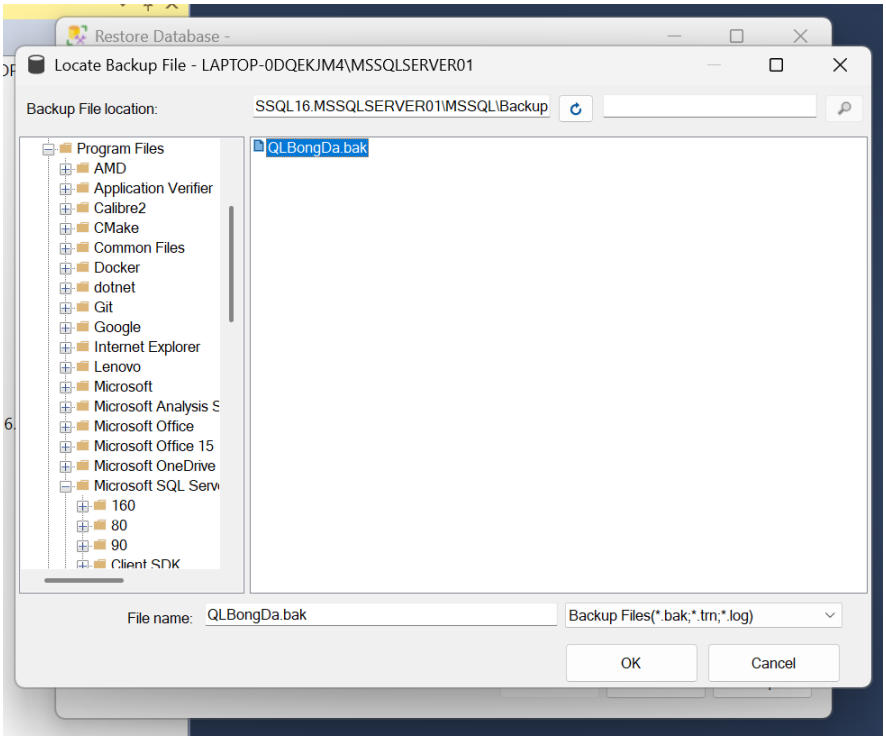


7.3 Restore database

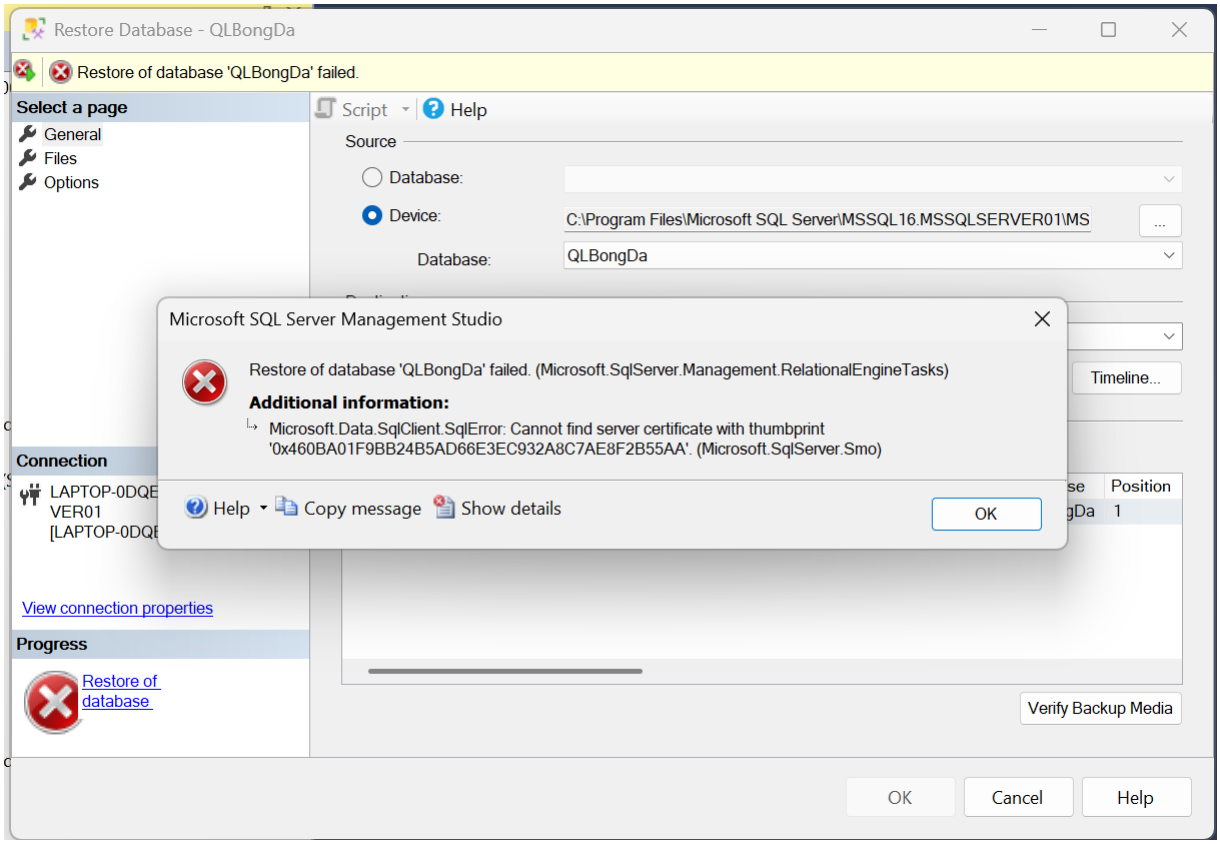
- Chuột phải vào mục **Databases** → chọn **Restore Database...**



- Chọn file `QLBongDa.bak` đã copy từ Server A.



- SQL Server tự động hiển thị tên logical file, đường dẫn file đích.
- Nhấn **OK** để bắt đầu restore.
- Kết quả: Xuất hiện lỗi không thể restore

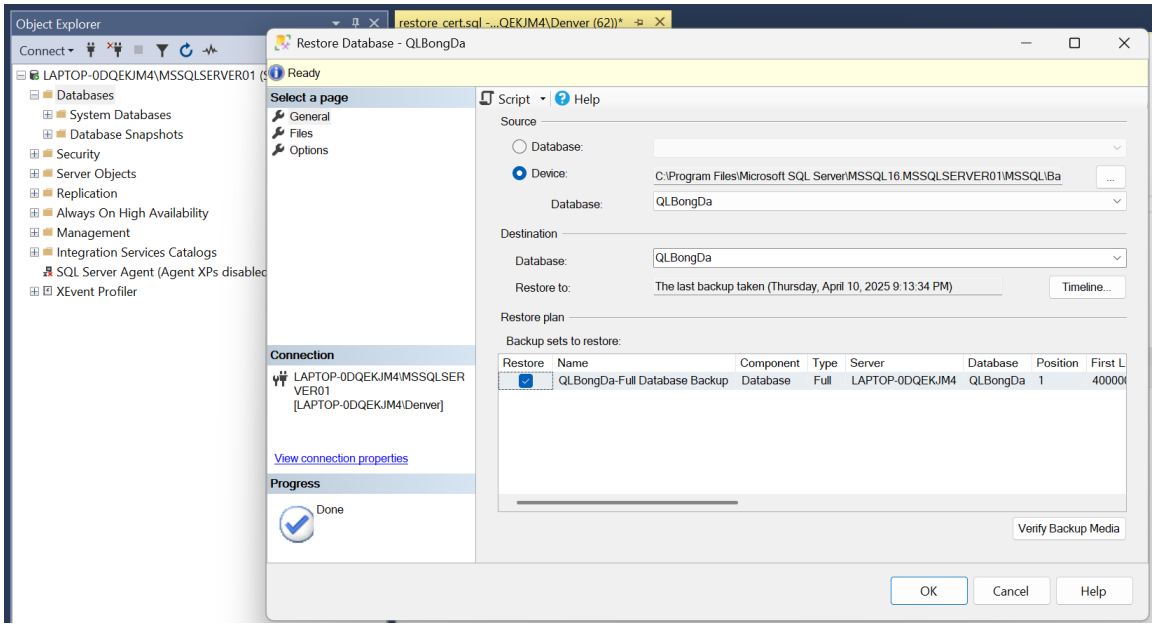


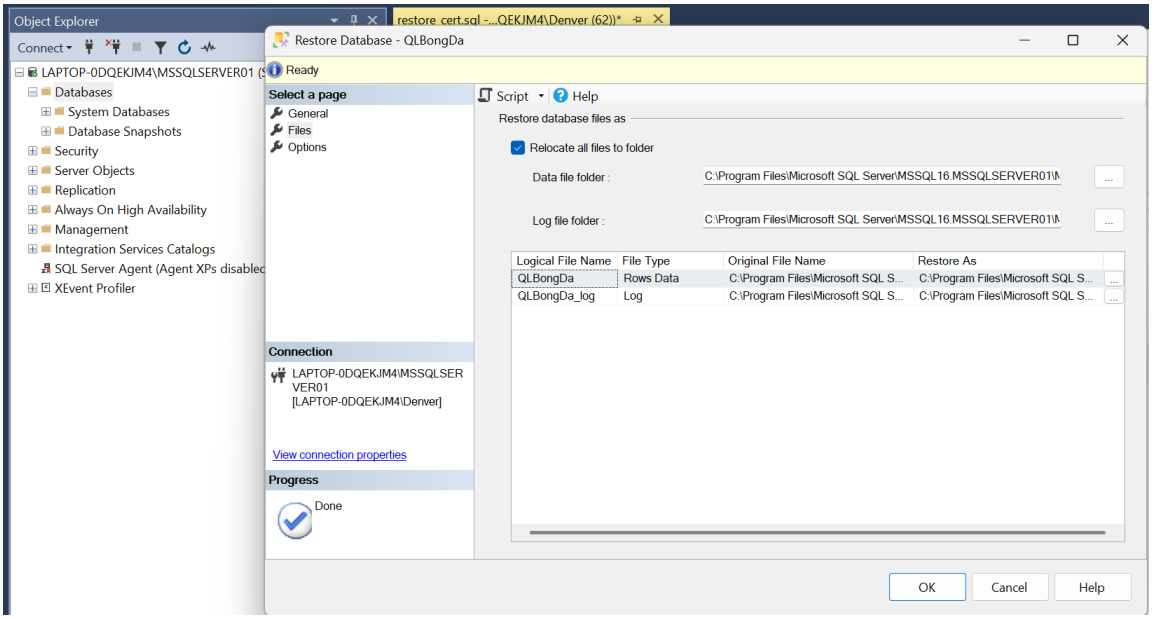
7.4 Nhận xét kết quả

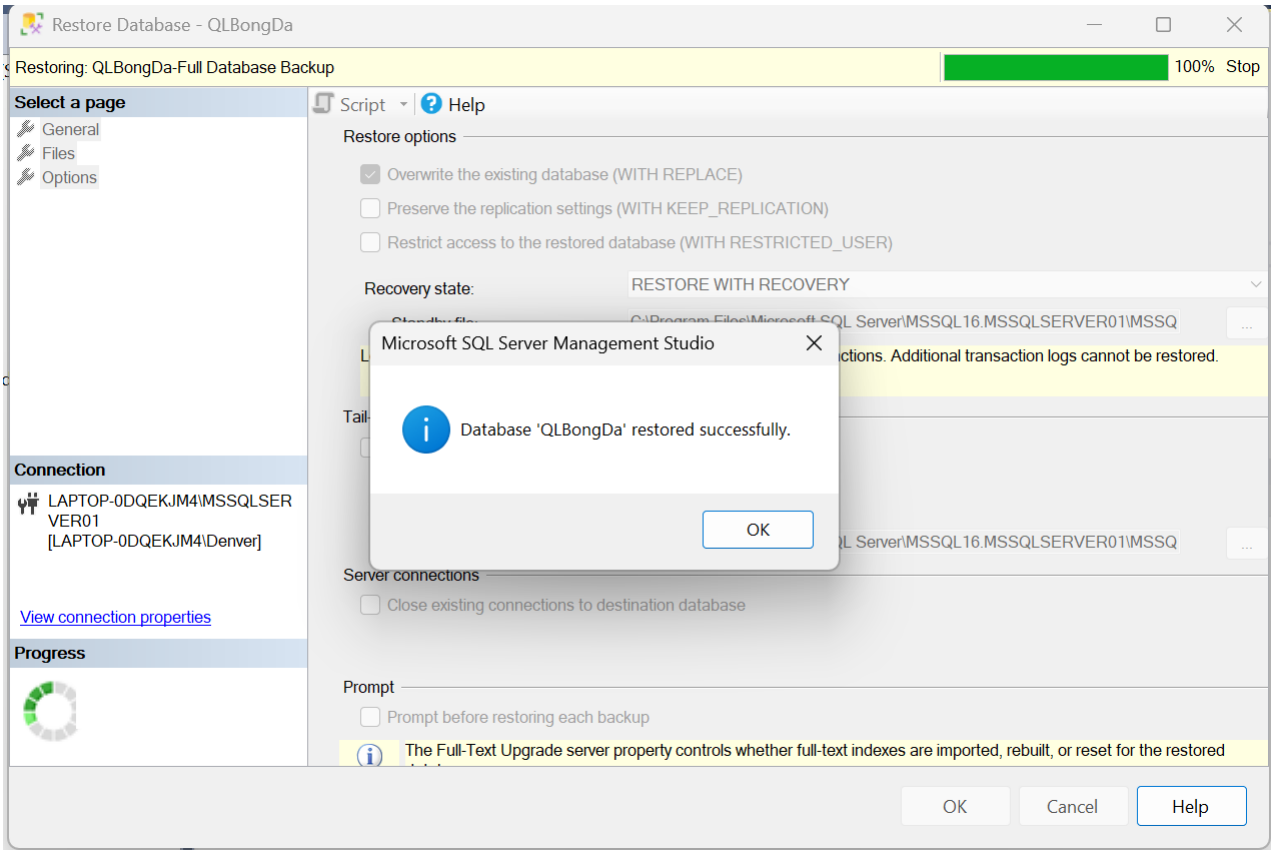
- Khi Backup - Restore database có sử dụng TDE, nếu không thực hiện backup và restore certificate từ Server A sang Server B, thao tác restore sẽ **thất bại**.
- Lỗi hiển thị do SQL Server không tìm thấy certificate tương ứng để giải mã CSDL.

8 Xử lý lỗi Backup - Restore khi sử dụng TDE

Thực hiện xử lý lỗi tương tự như câu e để Backup và Restore certificate từ server A sang server B. Sau khi thực hiện các bước trên thì có thể Backup - Restore CSDL một cách bình thường:







8.1 Kết luận

- Certificate là thành phần **bắt buộc** khi restore CSDL đã mã hóa bằng TDE.
- Nếu không khôi phục certificate đúng cách, việc restore sẽ luôn thất bại.
- Sau khi restore thành công, CSDL vẫn được mã hóa như ban đầu.