

TRƯỜNG ĐẠI HỌC TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
TRUNG TÂM PHÁT TRIỂN CÔNG NGHỆ THÔNG TIN

---

## BÀI TẬP THỰC HÀNH 03

An Ninh Thông Tin



Mã Môn: IE103  
Tên Môn: Quản Lý Thông Tin  
Lớp: LT.K2025.2.CNTT  
Giảng Viên: Ths. Nguyễn Thành Luân  
Thực Hiện: Nhóm 02

Tháng 12, 2025

# Mục Lục

---

Trang Bìa (Bài Tập Thực Hành 03) .....	i
Mục Lục .....	ii
Danh Sách Bảng .....	ii
Danh Sách Hình Ảnh .....	ii
Danh Sách Mã Nguồn .....	iii
Danh Sách TODO .....	iii
Nhóm 02 .....	iv
1. Bài 1 .....	1
1.1. A. TỔ CHỨC DỮ LIỆU .....	1
1.2. B. AN TOÀN DỮ LIỆU .....	3
2. Bài 2 .....	12
2.1. A. IMPORT VÀ EXPORT .....	12
2.2. B. XÁC THỰC NGƯỜI DÙNG .....	16
2.3. C. PHÂN QUYỀN NGƯỜI DÙNG .....	21

## DANH SÁCH BẢNG

---

Nhóm 02 .....	iv
---------------	----

## DANH SÁCH HÌNH ẢNH

---

Figure 1	Tasks > Back Up...	3
Figure 2	Destination: Chọn Disk .....	4
Figure 3	Backup Completed Successfully .....	4
Figure 4	Xóa DB .....	5
Figure 5	Restore (Khôi Phục) .....	5
Figure 6	Restore (Khôi Phục) Hoàn Thành .....	6
Figure 7	SQL: Sao Lưu .....	7
Figure 8	SQL: Xóa DB .....	7
Figure 9	SQL: Khôi Phục .....	8
Figure 10	SQL Server Log .....	8

Figure 11	Ý nghĩa các trường thông tin trong Log .....	9
Figure 12	Mã Hóa Dữ Liệu .....	11
Figure 13	Import/Export > Import Data from File(s)... .....	12
Figure 14	Chọn lựa file Excel chứa dữ liệu. ....	13
Figure 15	Các trường (field) và kiểu dữ liệu (data type) .....	13
Figure 16	Kết quả Import .....	14
Figure 17	Import/Export > Export Data to File .....	14
Figure 18	Chỉ định thể loại file và nơi lưu trữ .....	15
Figure 19	Thông báo EXPORT thành công. ....	15
Figure 20	Kiểm tra lại kết quả EXPORT .....	16

## DANH SÁCH MÃ NGUỒN

---

## DANH SÁCH TODO

---

TODO 1	Cách 2: SQL - SQL Script .....	6
TODO 2	Bài 2.B: SQL Script .....	16
TODO 3	Bài 2.C: SQL Script .....	21

## Nhóm 02

---

Thành viên của nhóm và bảng nhiệm vụ cho Bài Tập Thực Hành 03.

No	MSSV	Họ và Tên	Phụ Trách
1	25410291	Đinh Xuân Sâm	• Tổng hợp & Biên tập.
2	25410319	Đặng Hữu Toàn	
3	25410321	Nguyễn Điền Triết	
4	25410204	Trương Xuân Hậu	• Bài 2
5	25410338	Lê Anh Vũ	
6	25410176	Trần Sơn Bình	• Bài 1
7	25410247	Lê Kim Long	
8	25410337	La Anh Vũ	
9	25410209	Lê Ngọc Hiệp	
10	25410271	Nguyễn Thị Ngọc Nhung	

Nhóm 02

# BÀI 1

---

Tìm hiểu các vấn đề sau trên HQT CSDL SQLServer.

## 1.1. A. TỔ CHỨC DỮ LIỆU

### 1.1.1. Kiểu dữ liệu (data type) và cách sử dụng

Các kiểu dữ liệu (data type) dùng cho các trường (field) và cách sử dụng chúng.

- **Kiểu số (Exact Numerics):**
  - **INT** : Lưu số nguyên (4 bytes). Dùng cho ID, số lượng.
  - **BIGINT** : Số nguyên lớn (8 bytes). Dùng khi dữ liệu vượt quá 2 tỷ.
  - **BIT** : Giá trị **0** hoặc **1** (tương tự Boolean). Dùng cho trạng thái (Đúng/Sai).
  - **DECIMAL(p,s)** / **NUMERIC** : Số thực chính xác. Dùng cho tiền tệ, tính toán khoa học.
- **Kiểu chuỗi ký tự (Character Strings):**
  - **VARCHAR(n)** : Chuỗi ký tự độ dài thay đổi (không dấu). Dùng cho tên tiếng Anh, mã số.
  - **NVARCHAR(n)** : Chuỗi ký tự Unicode (có dấu tiếng Việt). Lưu ý: Tốn gấp đôi dung lượng so với **VARCHAR**.
  - **CHAR(n)** : Chuỗi ký tự độ dài cố định. Dùng cho các mã có độ dài cố định (VD: Mã bưu điện).
- **Kiểu ngày giờ (Date and Time):**
  - **DATETIME** : Ngày và giờ chính xác đến mili-giây.
  - **DATE** : Chỉ lưu ngày (YYYY-MM-DD).

### 1.1.2. Dung lượng (KB) tối đa của 1 row

Dung lượng (KB) tối đa của 1 row trong 1 table mà SQL Server cho phép.

- Dung lượng tối đa của một dòng (row) trong SQL Server được giới hạn bởi kích thước của một trang dữ liệu (Page).
- Giới hạn này là **8.060 bytes** (xấp xỉ 8KB).

### 1.1.3. Dung lượng (KB) tối đa của 1 table

Dung lượng (KB) tối đa của 1 table mà SQL Server cho phép.

- Về lý thuyết, kích thước bảng chỉ bị giới hạn bởi dung lượng lưu trữ của hệ điều hành (File System) và phiên bản SQL Server.
- Dung lượng tối đa của Database là **524,272 Terabytes**.
  - Do đó, 1 table có thể chứa lượng dữ liệu khổng lồ gần như không giới hạn trong thực tế, miễn là lưu trữ còn chỗ trống và xử lý được vấn đề hiệu năng.

#### 1.1.4. Ý nghĩa các table hệ thống (System tables)

Tìm hiểu ý nghĩa các table hệ thống (System tables) trong CSDL Master như: `sysusers` ; `syssservers` ; `sysxlogin` .

- `sysusers` :
  - Chứa thông tin về mỗi người dùng (user) hoặc nhóm (role) được phép truy cập trong cơ sở dữ liệu.
  - Đổi thành `sys.database_principals` (Transact-SQL) ở các phiên bản SQL Server hiện đại.
- `syssservers` :
  - Chứa thông tin về các server liên kết (Linked Servers) hoặc remote servers mà SQL Server này có thể giao tiếp.
  - Đổi thành `sys.servers` (Transact-SQL) ở các phiên bản SQL Server hiện đại.
- `sysxlogins` :
  - Chứa thông tin về tài khoản đăng nhập (Login account) và mật khẩu (đã mã hóa) cấp server.
  - Hoặc `syslogins`
  - Hiện nay là `sys.server_principals` hoặc `sys.sql_logins`.

Tham khảo: [Mapping System Tables to System Views \(Transact-SQL\)](#)

#### 1.1.5. Các file của một Cơ Sở Dữ Liệu (DB)

Khi người dùng tạo 1 CSDL, SQL Server yêu cầu tạo ra tối thiểu bao nhiêu file? Ý nghĩa mỗi file? Chú ý câu lệnh:

```
1 CREATE DATABASE AAA;
```

Khi tạo một CSDL (lệnh CREATE DATABASE AAA), SQL Server yêu cầu tạo ra tối thiểu 2 file:

##### 1. File dữ liệu chính (Primary Data File - đuôi `.mdf`):

- Ý nghĩa: Chứa thông tin khởi tạo database và lưu trữ dữ liệu chính của các bảng, view, index...
- Mỗi database bắt buộc phải có 1 file này.

##### 2. File nhật ký giao dịch (Transaction Log File - đuôi `.ldf`):

- Ý nghĩa: Ghi lại tất cả các giao dịch ( `insert` , `update` , `delete` ) và các sửa đổi CSDL.
- Dùng để khôi phục dữ liệu (restore) khi có sự cố, đảm bảo tính toàn vẹn (ACID).

Ngoài ra có thể có Secondary Data Files ( `.ndf` ) nếu cần chia nhỏ ra (optional).

### 1.1.6. Số user có thể connect cùng 1 thời điểm là bao nhiêu?

Số lượng kết nối tối đa (user connections) mà SQL Server hỗ trợ là **32.767** kết nối ([Server configuration: user connections](#)).

- “SQL Server allows a maximum of 32,767 user connections.”

## 1.2. B. AN TOÀN DỮ LIỆU

### 1.2.1. 7. Backup (Sao Lưu) và Restore (Khôi Phục)

Hãy backup CSDL AAA thành 1 file AAA.BAK, sau đó xóa CSDL AAA và hãy khôi phục AAA nhờ vào AAA.BAK. Thực hiện bằng 2 cách:

- Cách 1: Dùng giao diện.
  - Yêu cầu: Chụp màn hình các bước thực hiện (Chụp đúng phần cần thiết và rõ nét, có thể dùng công cụ Snipping Tool).
- Cách 2: Dùng câu lệnh.
  - Yêu cầu: Dùng các lệnh SQL để thực hiện.

#### 1.2.1.1. Cách 1: Giao Diện (SSMS)

##### 1.2.1.1.1. Backup (Sao Lưu)

- Click chuột phải vào CSDL AAA > **Tasks** > **Back Up...**

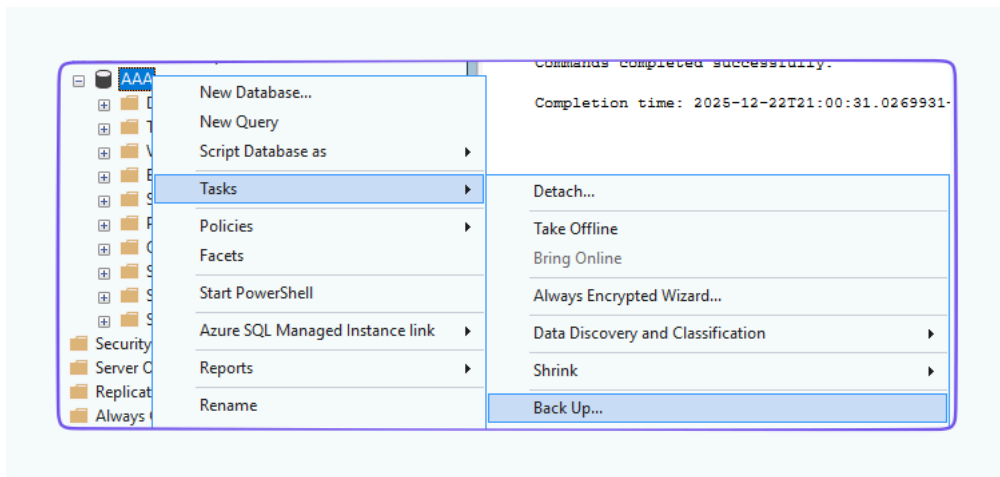


Figure 1: Tasks > Back Up...

- Mục **Destination**: Chọn **Disk**, chọn đường dẫn lưu file AAA.BAK.

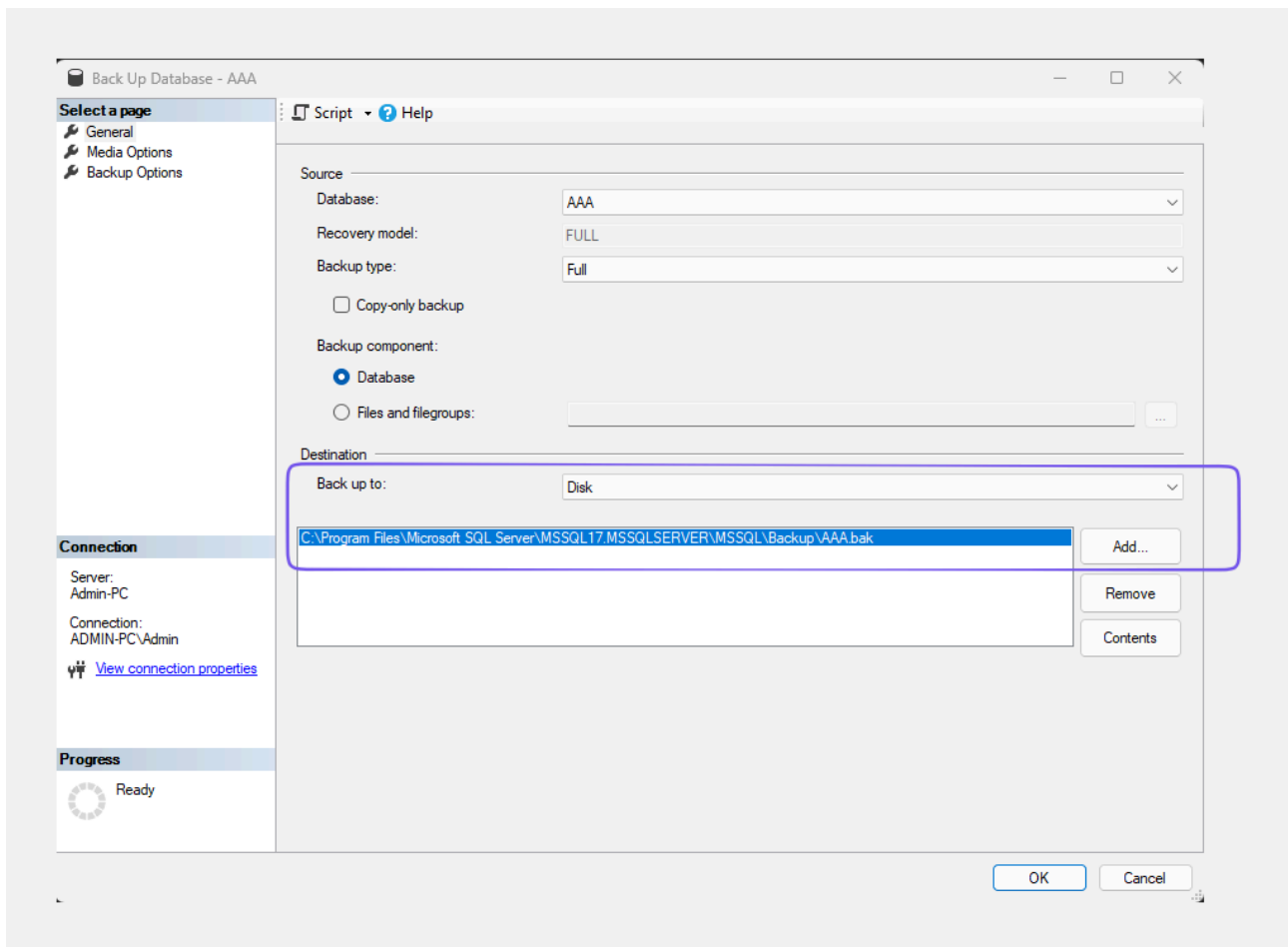


Figure 2: Destination: Chọn Disk

- Nhấn **OK**.
  - Màn hình thông báo “The backup of database ‘AAA’ completed successfully”).

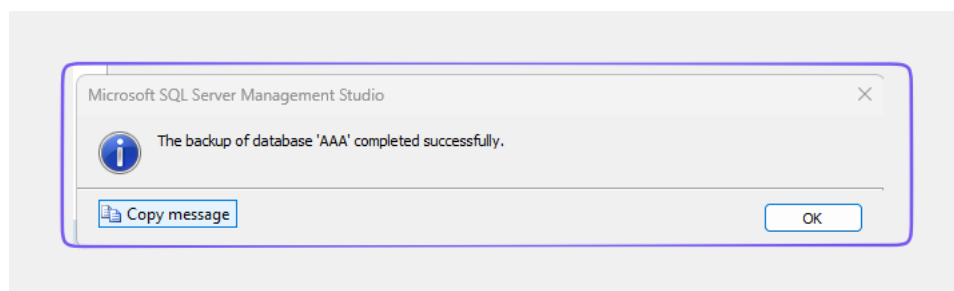


Figure 3: Backup Completed Successfully

#### 1.2.1.1.2. Xóa DB

- Click chuột phải vào CSDL AAA -> **Delete**.
- Check vào ô **Close existing connections** (để ngắt các kết nối đang chạy).



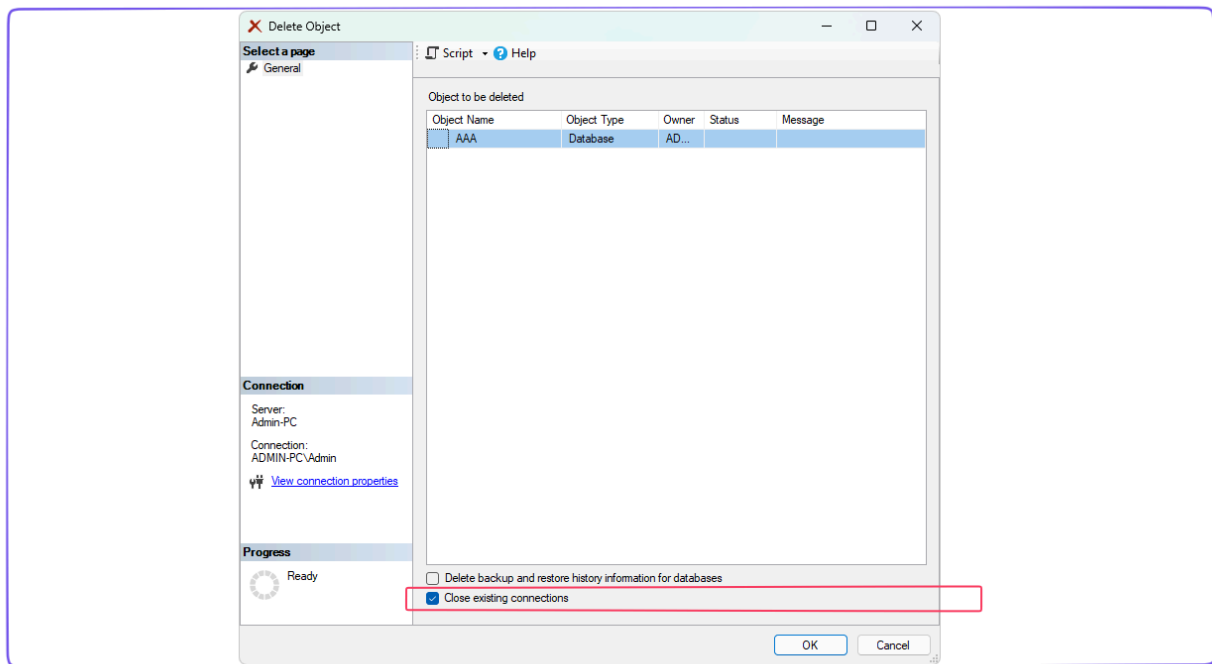


Figure 4: Xóa DB

- Nhấn **OK**.

#### 1.2.1.1.3. Restore (Khôi Phục)

- Click chuột phải vào thư mục **Databases** -> **Restore Database...**
- Chọn **Database** -> Nhấn nút **...** để duyệt file -> Chọn file **AAA.BAK** vừa tạo.

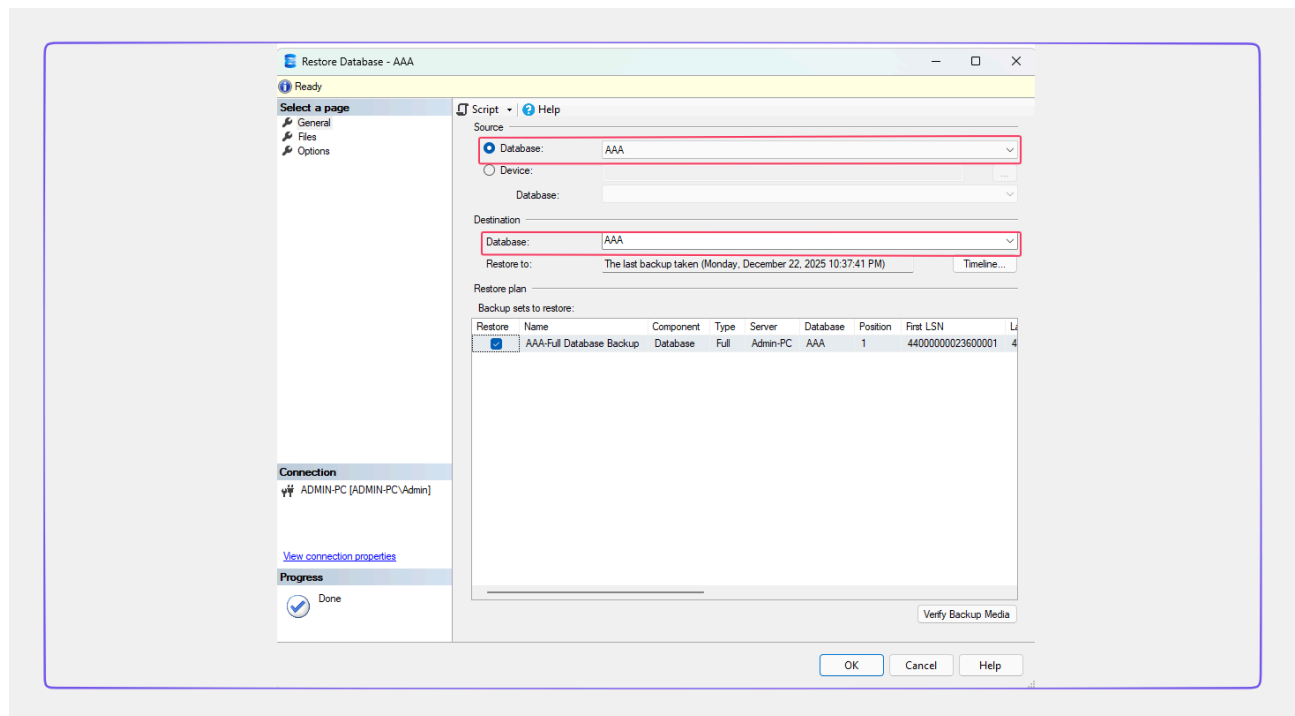


Figure 5: Restore (Khôi Phục)

- Nhấn **OK**.

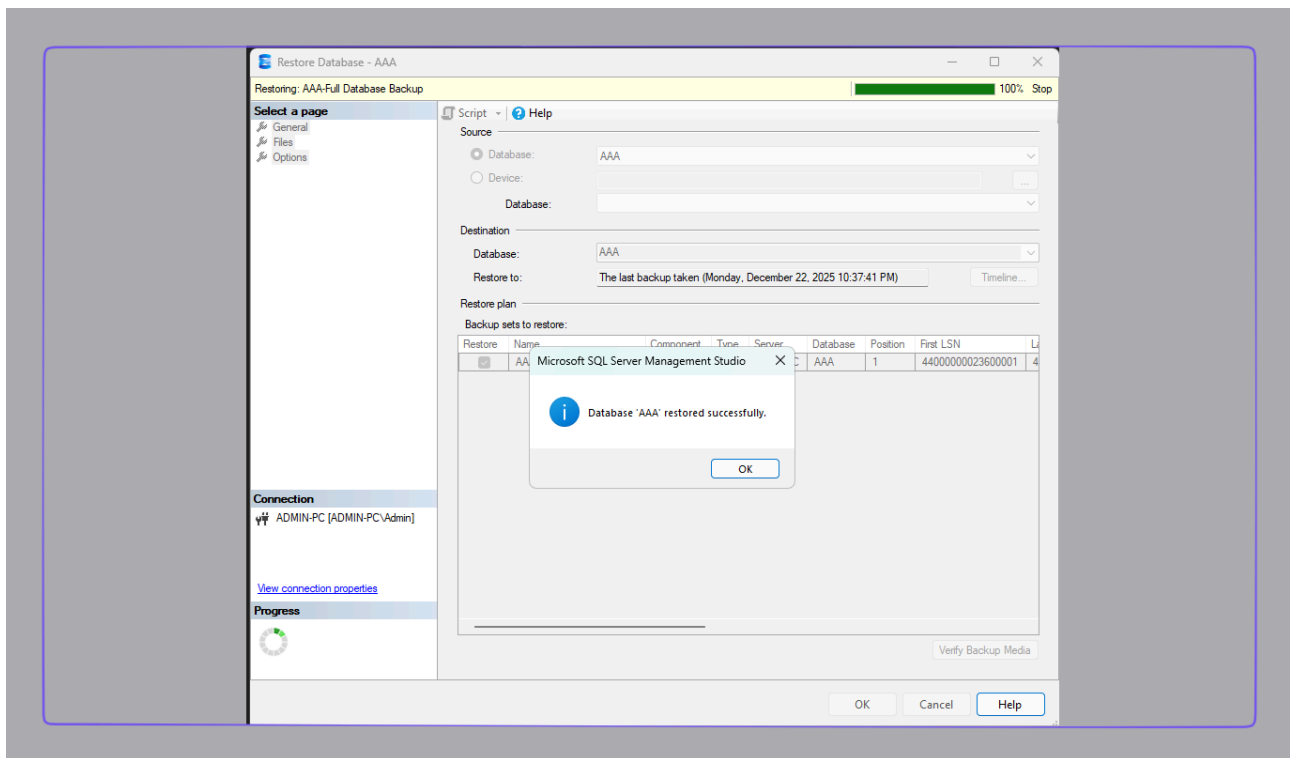


Figure 6: Restore (Khôi Phục) Hoàn Thành

#### 1.2.1.2. Cách 2: SQL

**TODO:** Cách 2: SQL - SQL Script

Mở cửa sổ **New Query** và chạy lần lượt các đoạn lệnh.

##### 1.2.1.2.1. Backup (Sao Lưu)

```
1 BACKUP DATABASE AAA
2 TO DISK = 'C:\Users\Admin\Desktop\Backup\AAA.BAK'
3 WITH FORMAT;
4 GO
```

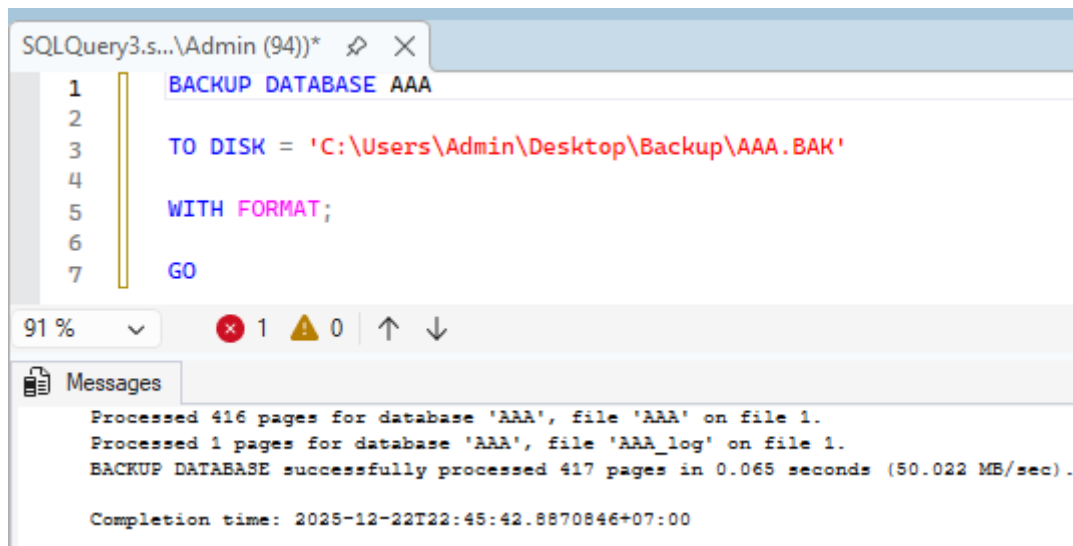


Figure 7: SQL: Sao Lưu

#### 1.2.1.2.2. Xóa DB

```

1  ALTER DATABASE AAA SET SINGLE_USER WITH ROLLBACK IMMEDIATE; -- để đảm bảo không có ai đang
2  dùng Database
   DROP DATABASE AAA;

```

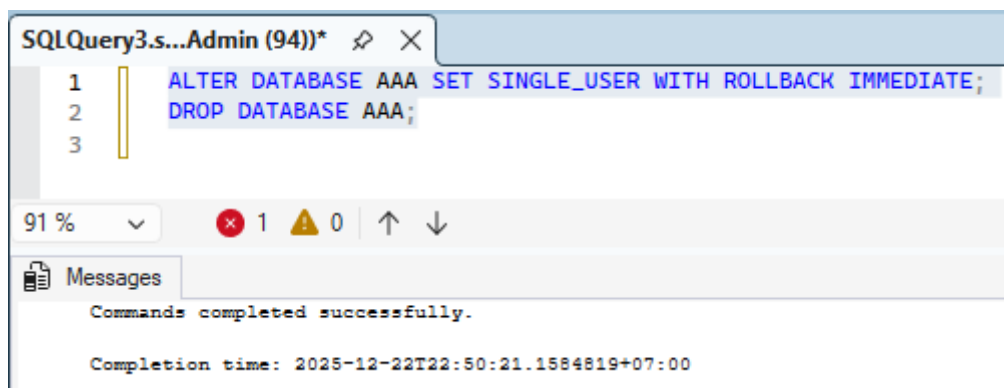


Figure 8: SQL: Xóa DB

#### 1.2.1.2.3. Restore (Khôi Phục)

```

1  RESTORE DATABASE AAA
2  FROM DISK = 'C:\Users\Admin\Desktop\Backup\AAA.BAK;
3  GO

```

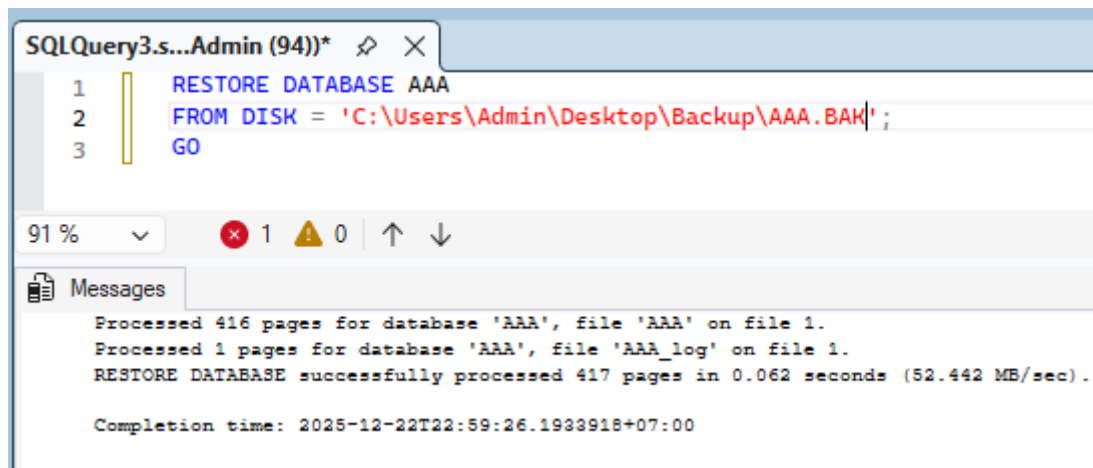


Figure 9: SQL: Khôi Phục

### 1.2.2. SQL Server Log

Hãy đọc SQL Server Log trong phần SQL Enterprise → Management → SQL Server Log. Trình bày màn hình các bước thực hiện và ý nghĩa của mỗi trường thông tin trong bảng Log.

- **Đường dẫn:** Trong SQL Server Management Studio (SSMS), mở rộng mục **Management** -> Mở rộng **SQL Server Logs**. Click đúp vào **Current** (hoặc Archive #1, #2).

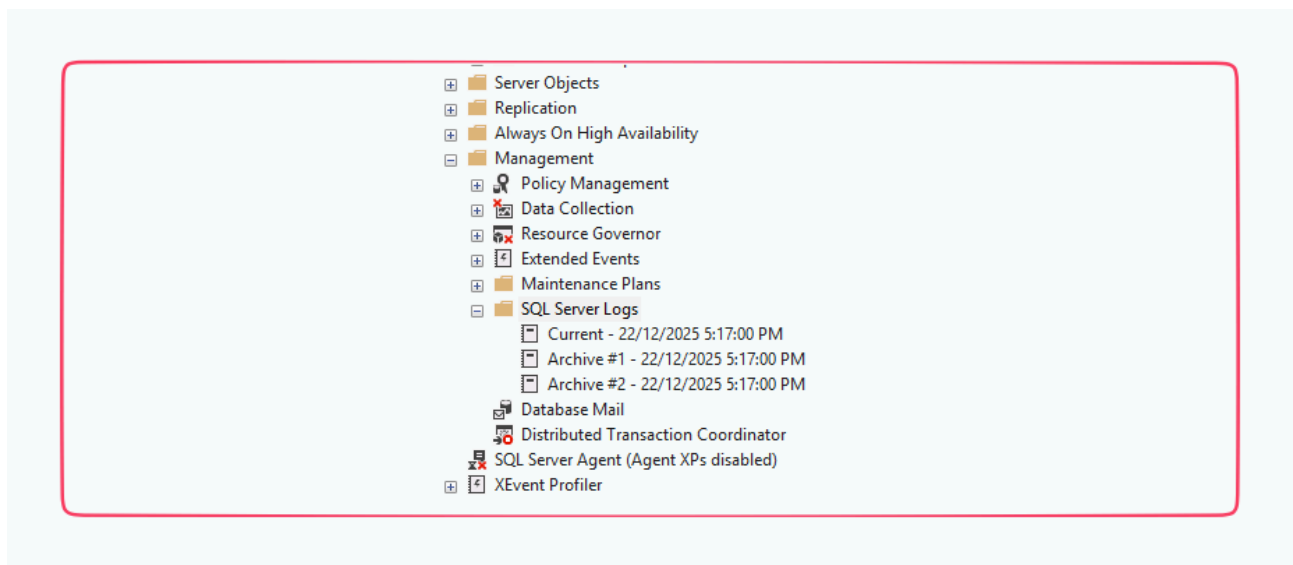


Figure 10: SQL Server Log

- Ý nghĩa các trường thông tin trong Log:
  - **Date:** Thời gian (ngày/giờ) sự kiện xảy ra.
  - **Source:** Nguồn gốc của thông báo (thường là Server, Logon, hoặc spid - Session Process ID).
  - **Message:** Nội dung chi tiết của thông báo (“Database backed up...”, “Login failed for user...”).

- Mục đích: Giúp quản trị viên theo dõi sức khỏe hệ thống, phát hiện lỗi đăng nhập, kiểm tra các tác vụ backup thành công hay thất bại.

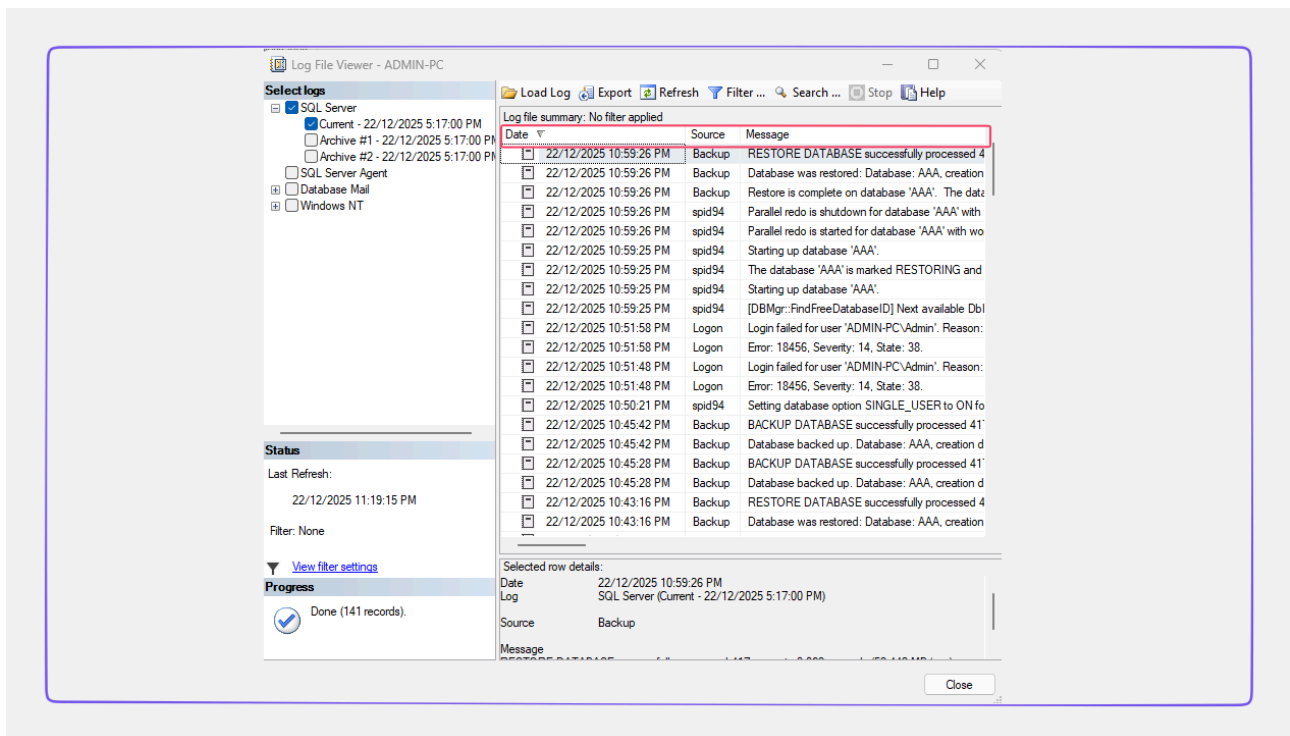


Figure 11: Ý nghĩa các trường thông tin trong Log

### 1.2.3. Employees

Cho bảng `Employees` có cấu trúc như sau:

```

1 CREATE TABLE Employees (
2     EmployeeID INT PRIMARY KEY,
3     FirstName VARCHAR (50) NOT NULL,
4     LastName VARCHAR (50) NOT NULL,
5     BirthDate DATE NOT NULL,
6     HireDate DATE NOT NULL );
7 GO

```

Giả sử tạo 1 View tên là `EmployeeNames` như sau:

```

1 CREATE VIEW EmployeeNames
2 AS
3     SELECT FirstName, LastName
4     FROM Employees

```

Hỏi câu lệnh `INSERT` dưới đây có thực hiện được hay không? Tại sao?

```

1 INSERT INTO EmployeeNames (FirstName, LastName) VALUES ('QuanLyThongTin', 'IE103');

```

### Trả lời:

- Câu lệnh INSERT này KHÔNG thực hiện được.

### Giải thích:

- **Cấu trúc bảng ( Employees ):**
  - Cột BirthDate được định nghĩa là NOT NULL (Bắt buộc có dữ liệu).
  - Cột HireDate được định nghĩa là NOT NULL (Bắt buộc có dữ liệu).
  - Không có giá trị mặc định ( DEFAULT ) nào được khai báo cho 2 cột này.
- **Cấu trúc View ( EmployeeNames ):**
  - Chỉ chứa 2 cột: FirstName và LastName .
- **Hành động INSERT:**
  - Câu lệnh: INSERT INTO EmployeeNames (FirstName, LastName) VALUES (...)
  - Khi insert vào View, SQL Server thực chất sẽ cố gắng insert vào bảng gốc Employees .
  - Câu lệnh trên cung cấp FirstName , LastName (và EmployeeID nếu ID tự tăng - Identity, trong trường hợp này thì không).
  - Tuy nhiên, nó không cung cấp giá trị cho BirthDate và HireDate .

### Thực nghiệm:

- Nhận thông báo lỗi.

```
1 Msg 515, Level 16, State 2, Line 15
2 Cannot insert the value NULL into column 'HireDate', table 'BTTH2.dbo.Employees'; column
  does not allow nulls. INSERT fails.
```

### Kết luận:

- Do bảng gốc yêu cầu BirthDate và HireDate không được để trống ( NOT NULL ), nhưng câu lệnh INSERT thông qua View lại không cung cấp giá trị cho chúng, nên SQL Server sẽ báo lỗi vi phạm ràng buộc dữ liệu.

#### 1.2.4. Mã Hóa Dữ Liệu

Cho hình bên dưới.

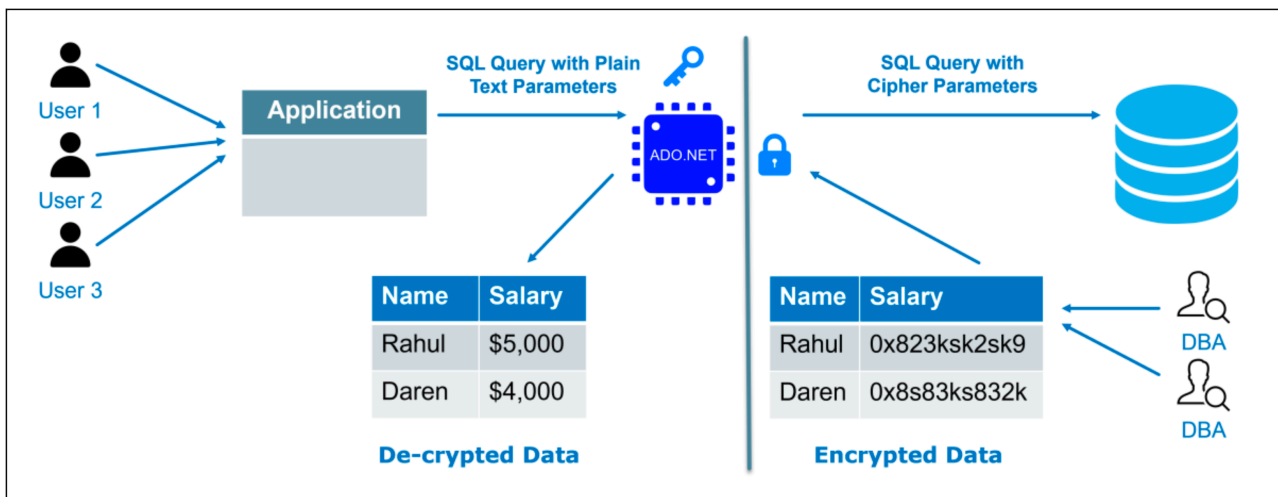


Figure 12: Mã Hóa Dữ Liệu

Hỏi hình trên đang mô tả mã hóa dữ liệu ở trạng thái nào? Vì sao?

Biết dữ liệu có 3 trạng thái là:

- Data at rest (trạng thái nghỉ),
- Data in use (trạng thái đang sử dụng),
- Data in motion/transit (trạng thái lưu chuyển).

Trả lời:

- Đồ họa trên miêu tả việc mã hóa dữ liệu ở các trạng thái sau đây.
- **Data in motion/transit (Trạng thái lưu chuyển)**
  - Mũi tên từ bộ phận điều khiển ADO.NET (phía ứng dụng/Client) gửi đến Database có ghi chú “SQL Query with Cipher Parameters” (Truy vấn SQL với các tham số mật mã).
  - ADO.NET: “a set of computer software components that programmers can use to access data and data services from a database.”
  - Điều này có nghĩa là dữ liệu đã được mã hóa ngay tại ứng dụng (Client-side) trước khi nó được gửi tới lưu trữ.
  - Do đó, trên đường truyền, dữ liệu hoàn toàn ở dạng mã hóa, bảo vệ nó khỏi việc bị nghe lén.
- **Data at rest (Trạng thái nghỉ)**
  - Ở phía bên phải (Database), hình ảnh biểu diễn dữ liệu ngay từ trước khi được đưa vào lưu trữ (hình trụ) là “Encrypted Data” (Dữ liệu đã mã hóa).
  - Ngay cả DBA (người quản trị cơ sở dữ liệu) khi truy xuất vào bảng cũng chỉ thấy chuỗi mã hóa vô nghĩa (ví dụ: 0x823ksk...).
  - Điều này chứng tỏ dữ liệu nằm trên thiết bị lưu trữ luôn ở trạng thái được mã hóa.

# BÀI 2

## 2.1. A. IMPORT VÀ EXPORT

### 2.1.1. Import Dữ Liệu Từ Excel File

Chọn 1 file dữ liệu (SV) từ excel, và import vào SQL Server.

- Chuột phải vào bảng cần IMPORT dữ liệu và chọn **Import/Export > Import Data from File(s)...**

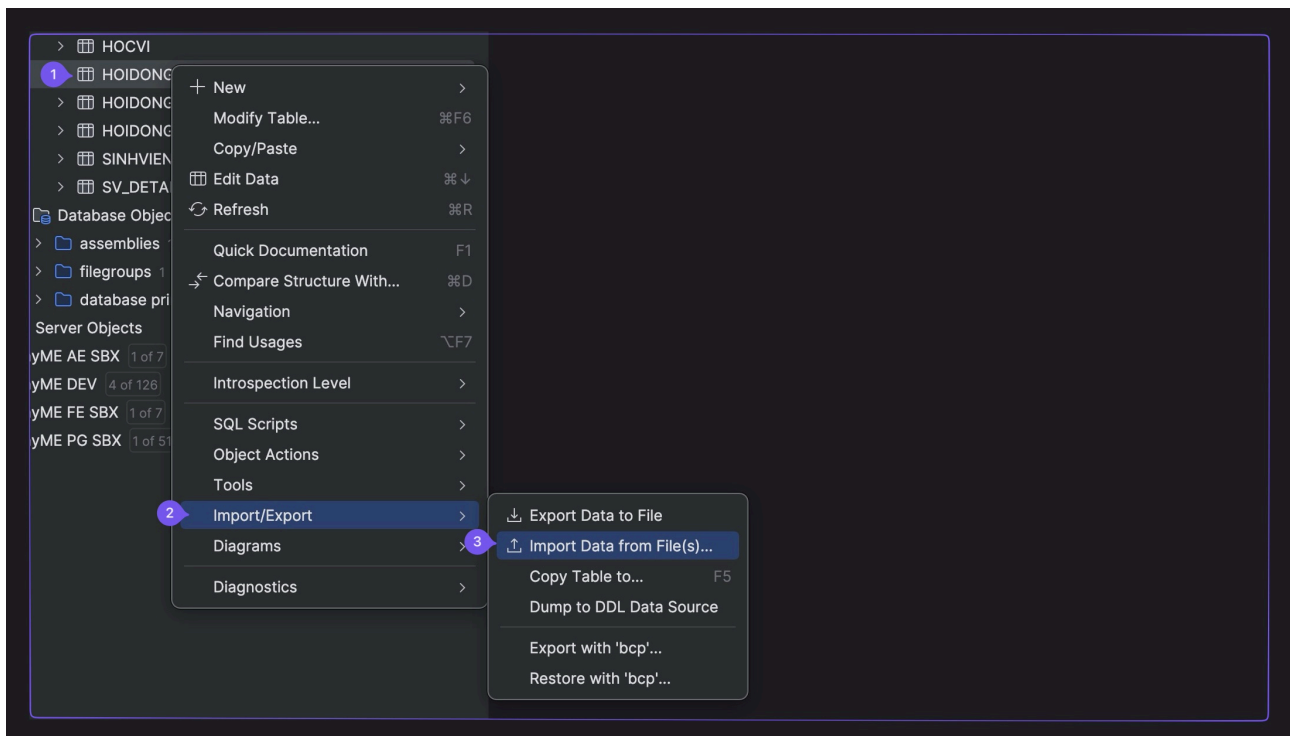


Figure 13: Import/Export > Import Data from File(s)...

- Chọn file Excel chứa dữ liệu.



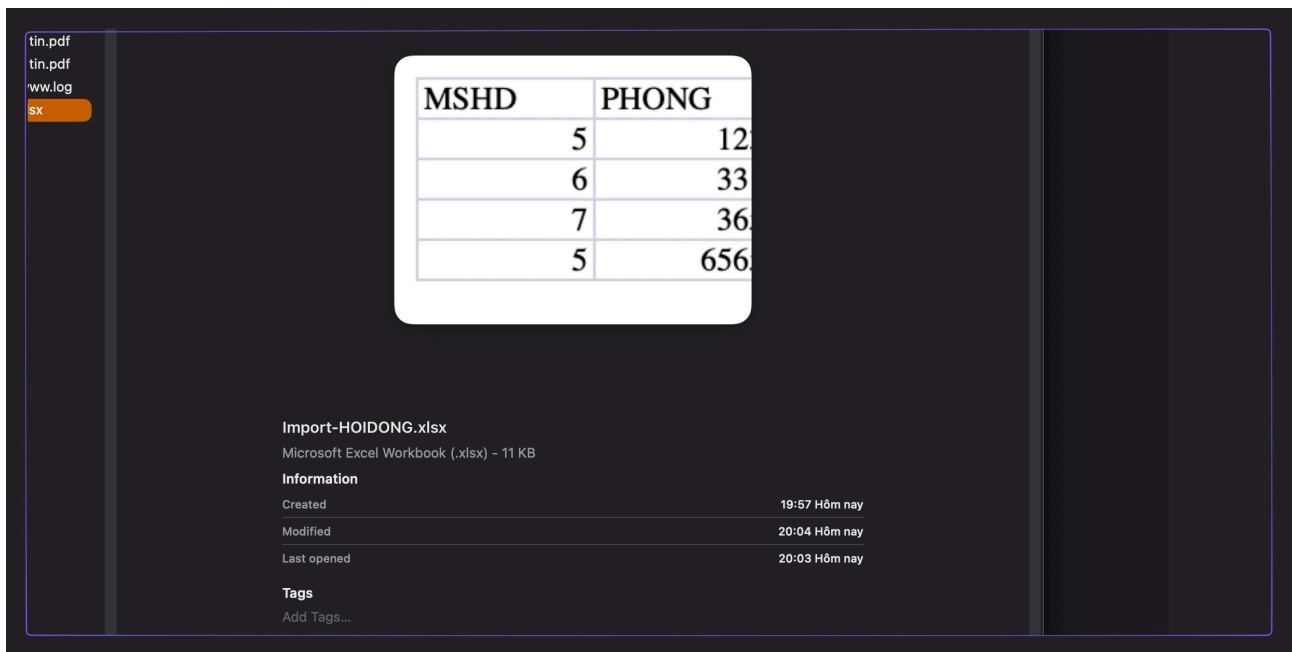


Figure 14: Chọn lựa file Excel chứa dữ liệu.

- Cấu hình các trường (field) và kiểu dữ liệu (data type) cho bảng.

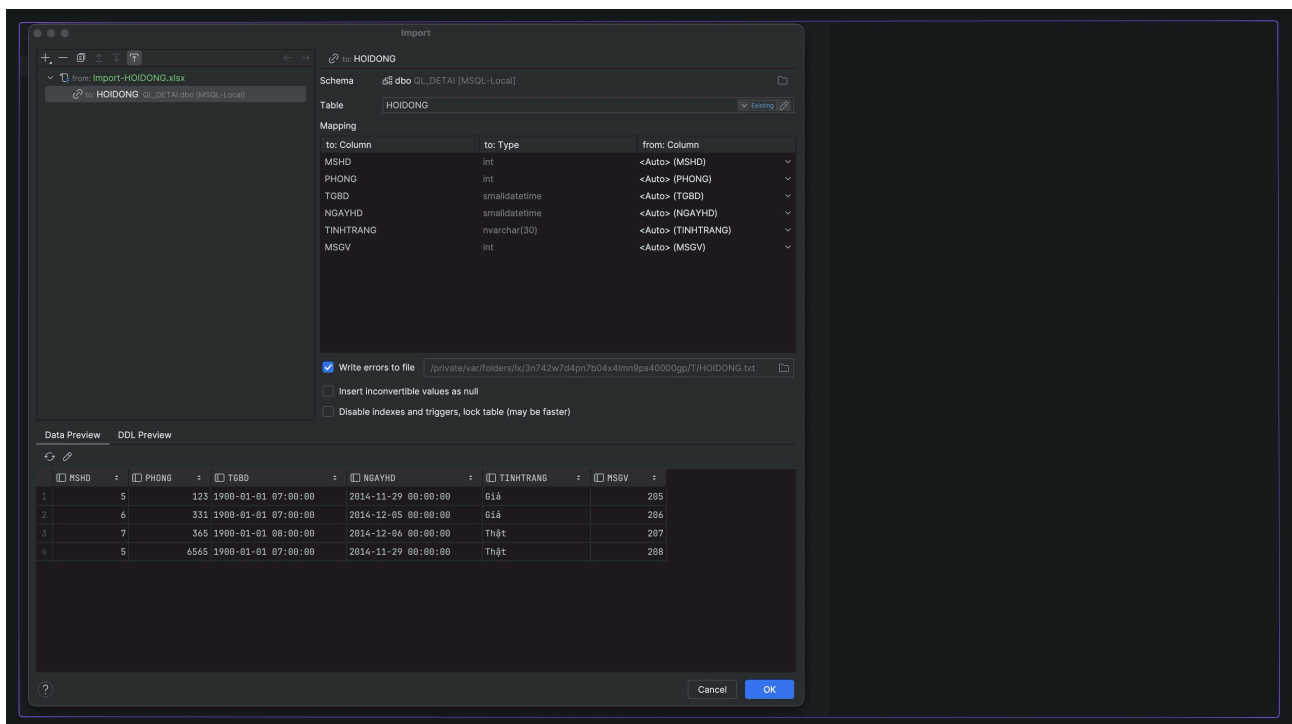


Figure 15: Các trường (field) và kiểu dữ liệu (data type)

- Kết quả: Dữ liệu đã được nhập (imported) thành công.

	MSKD	PHONG	TGBD	NGAYHD	TINHTRANG	MSGV
1	1	6969	1900-01-01 07:00:00	2014-11-29 00:00:00	Thật	201
2	2	6969	1900-01-01 07:00:00	2014-12-05 00:00:00	Thật	202
3	3	6969	1900-01-01 08:00:00	2014-12-06 00:00:00	Thật	203
4	4	2	1900-01-01 07:00:00	2014-11-29 00:00:00	Thật	201
5	5	123	1900-01-01 07:00:00	2014-11-29 00:00:00	Giả	205

Figure 16: Kết quả Import

### 2.1.2. Export Dữ Liệu tới Excel File

Chọn 1 table trong SQL Server, và export tới file Excel.

- Chuột phải vào bảng cần EXPORT dữ liệu và chọn **Import/Export > Export Data to File**

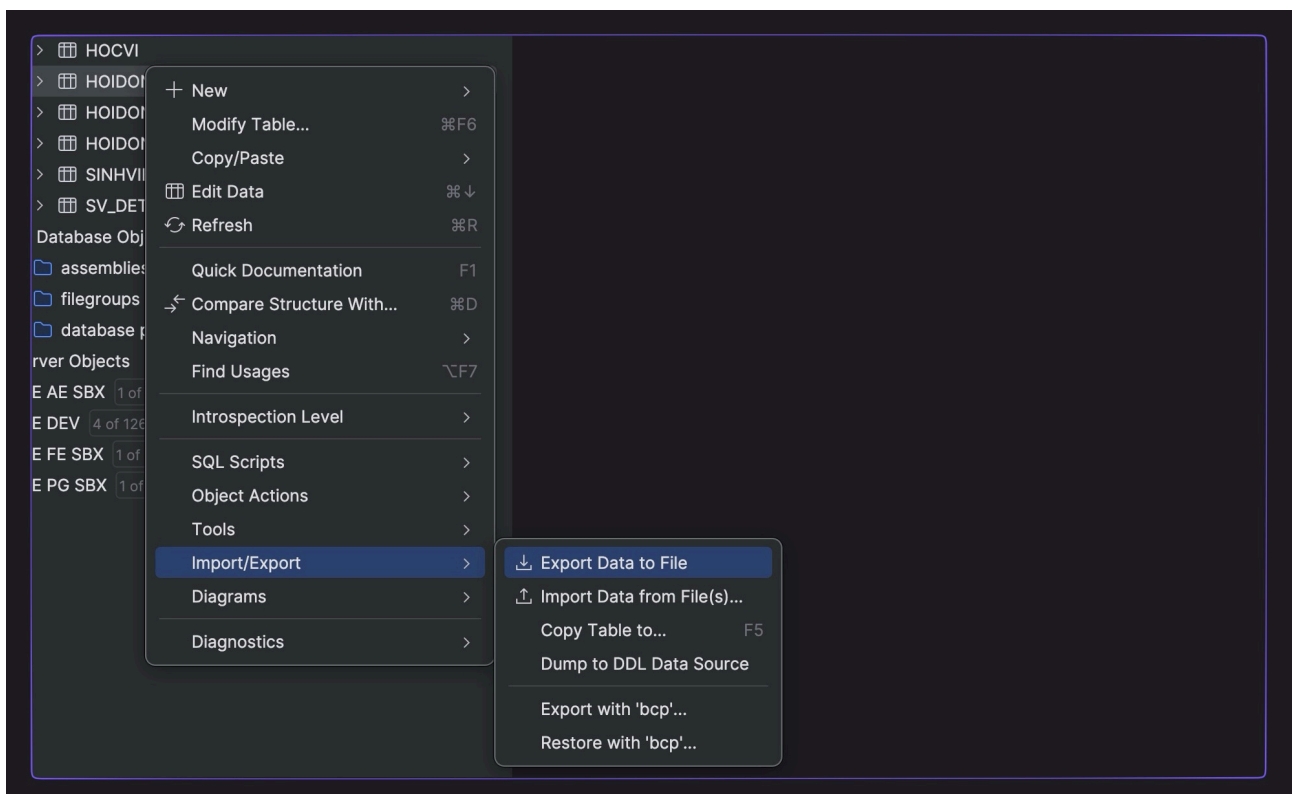


Figure 17: Import/Export > Export Data to File

- Chỉ định thể loại file (**Extractor**) và nơi lưu trữ (**Output file**).

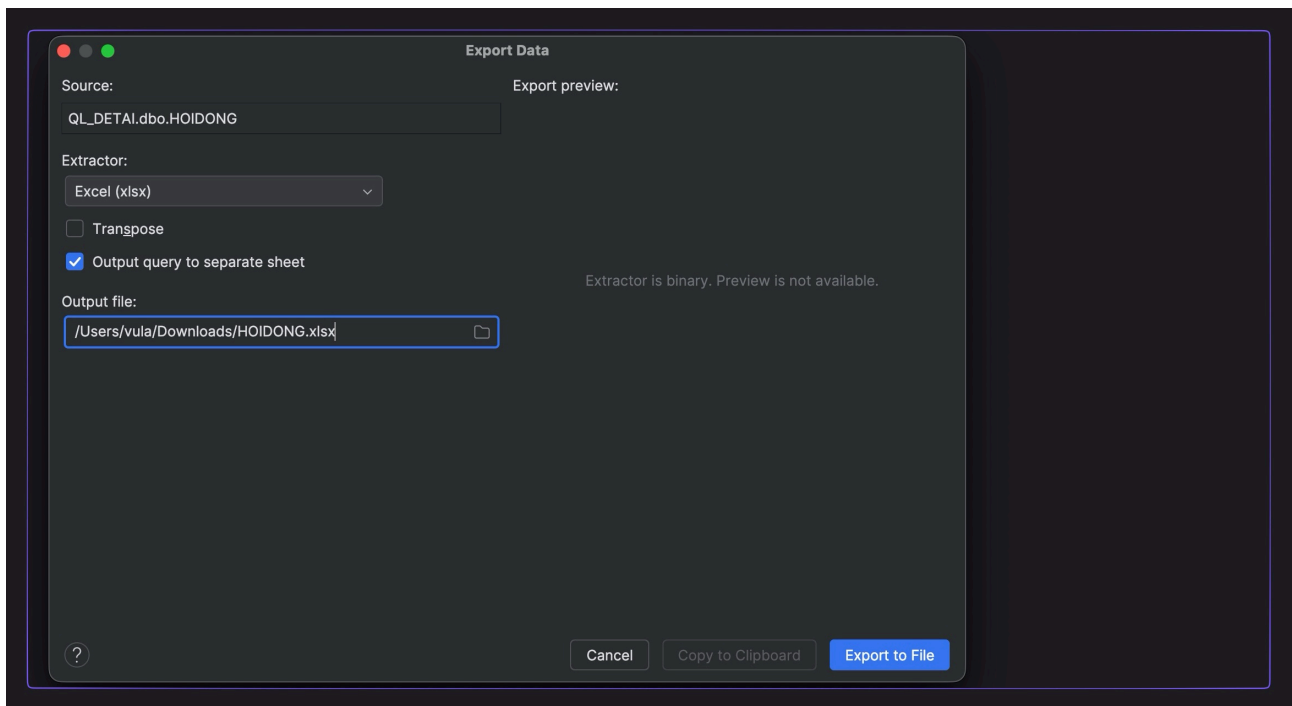


Figure 18: Chỉ định thể loại file và nơi lưu trữ

- Thông báo EXPORT thành công.

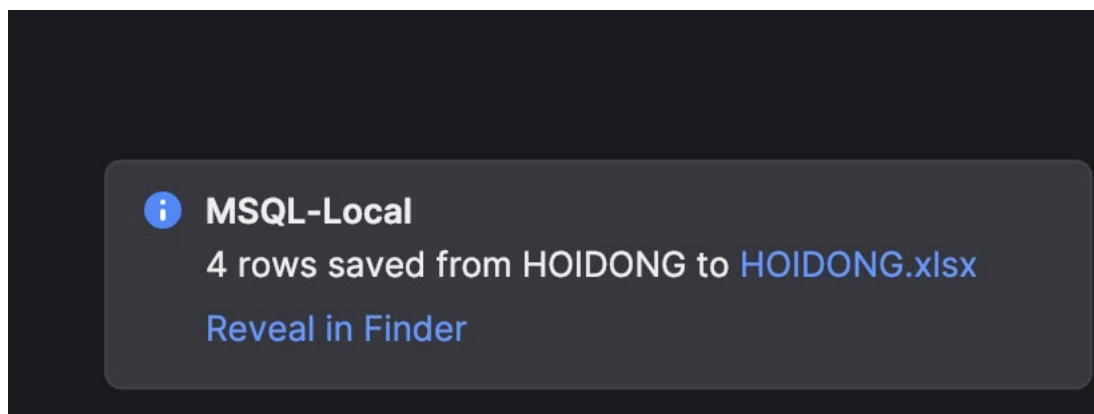


Figure 19: Thông báo EXPORT thành công.

- Kiểm tra lại kết quả EXPORT.

MSHD	PHONG	TGBD	NGAYHD	TINHTRANG	MSGV
1	6969	1900-01-0	2014-11-2	Thật	201
2	6969	1900-01-0	2014-12-0	Thật	202
3	6969	1900-01-0	2014-12-0	Thật	203
4	2	1900-01-0	2014-11-2	Thật	201

Figure 20: Kiểm tra lại kết quả EXPORT

## 2.2. B. XÁC THỰC NGƯỜI DÙNG

**TODO:** Bài 2.B: SQL Script

### 2.2.1. Mô tả yêu cầu

- Tạo 6 user từ `u1` đến `u6`
- Tạo 3 role từ `r1` đến `r3`
- Tạo nhóm:
  - `u1` thuộc `r1`;
  - `u2`, `u3` thuộc `r2`;
  - `u4`, `u5`, `u6` thuộc `r3`
- Thực hiện:
  - `r1` thành viên của `SysAdmin`
  - `r2` thành viên của `db_owner`, `db_accessadmin`
  - `r3` thành viên của `SysAdmin`, `db_owner`, `db_accessadmin`

### 2.2.2. B.1 Tạo Login L1 → L6

```

1 CREATE LOGIN L1 WITH PASSWORD = 'L1@12345678';
2 CREATE LOGIN L2 WITH PASSWORD = 'L2@12345678';
3 CREATE LOGIN L3 WITH PASSWORD = 'L3@12345678';
4 CREATE LOGIN L4 WITH PASSWORD = 'L4@12345678';
5 CREATE LOGIN L5 WITH PASSWORD = 'L5@12345678';
6 CREATE LOGIN L6 WITH PASSWORD = 'L6@12345678';

```

name	principal_id	type	type_desc	is_disabled
L1	262	S	SQL_LOGIN	0
L2	263	S	SQL_LOGIN	0
L3	264	S	SQL_LOGIN	0
L4	265	S	SQL_LOGIN	0
L5	266	S	SQL_LOGIN	0
L6	267	S	SQL_LOGIN	0

### 2.2.3. B.2 Tạo User U1 → U6 tương ứng với Login

```

1 CREATE USER U1 FOR LOGIN L1;
2 CREATE USER U2 FOR LOGIN L2;
3 CREATE USER U3 FOR LOGIN L3;
4 CREATE USER U4 FOR LOGIN L4;
5 CREATE USER U5 FOR LOGIN L5;
6 CREATE USER U6 FOR LOGIN L6;

```

uid	status	name	altuid	gid	hasdbac- cess	islogin	issqluser
7	0	U1	NULL	0	1	1	1
8	0	U2	NULL	0	1	1	1
9	0	U3	NULL	0	1	1	1
10	0	U4	NULL	0	1	1	1
11	0	U5	NULL	0	1	1	1
12	0	U6	NULL	0	1	1	1

### 2.2.4. B.3 Tạo Role r1, r2, r3

```

1 CREATE ROLE r1;
2 CREATE ROLE r2;
3 CREATE ROLE r3;

```

uid	status	name	altuid	gid	hasdbac- cess	islogin	issqluser
13	0	r1	1	13	0	0	0
14	0	r2	1	14	0	0	0
15	0	r3	1	15	0	0	0

## 2.2.5. B.4 Gán User vào Role

```
1 ALTER ROLE r1 ADD MEMBER U1;
2
3 ALTER ROLE r2 ADD MEMBER U2;
4 ALTER ROLE r2 ADD MEMBER U3;
5
6 ALTER ROLE r3 ADD MEMBER U4;
7 ALTER ROLE r3 ADD MEMBER U5;
8 ALTER ROLE r3 ADD MEMBER U6;
```

role_principal_id	Role	member_principal_id	User
13	r1	7	U1
14	r2	8	U2
14	r2	9	U3
15	r3	10	U4
15	r3	11	U5
15	r3	12	U6

## 2.2.6. B.5 Gán quyền hệ thống cho Role

### 2.2.6.1. r1 là thành viên của SysAdmin

```
1 ALTER SERVER ROLE sysadmin ADD MEMBER L1;
```

### 2.2.6.2. r2 là thành viên của db\_owner, db\_accessadmin

```
1 ALTER ROLE db_owner ADD MEMBER U2;
2 ALTER ROLE db_accessadmin ADD MEMBER U2;
3
4 ALTER ROLE db_owner ADD MEMBER U3;
5 ALTER ROLE db_accessadmin ADD MEMBER U3;
```

### 2.2.6.3. r3 là thành viên của SysAdmin, db\_owner, db\_accessadmin

```
1 ALTER SERVER ROLE sysadmin ADD MEMBER L4;
2 ALTER ROLE db_owner ADD MEMBER U4;
3 ALTER ROLE db_accessadmin ADD MEMBER U4;
4
5 ALTER SERVER ROLE sysadmin ADD MEMBER L5;
6 ALTER ROLE db_owner ADD MEMBER U5;
7 ALTER ROLE db_accessadmin ADD MEMBER U5;
8
9 ALTER SERVER ROLE sysadmin ADD MEMBER L6;
10 ALTER ROLE db_owner ADD MEMBER U6;
11 ALTER ROLE db_accessadmin ADD MEMBER U6;
```

### 2.2.7. Kiểm tra kết quả:

- Login thuộc sysadmin :

```
1 SELECT
2     ServerRole.principal_id AS [Role ID],
3     ServerRole.name AS [Server Role],
4     ServerLogin.principal_id AS [LOGIN ID],
5     ServerLogin.name AS [LOGIN Name],
6     ServerLogin.type_desc AS [Type]
7 FROM sys.server_role_members AS RoleMem
8 INNER JOIN sys.server_principals AS ServerRole
9     ON RoleMem.role_principal_id = ServerRole.principal_id
10 INNER JOIN sys.server_principals AS ServerLogin
11     ON RoleMem.member_principal_id = ServerLogin.principal_id
12 WHERE ServerRole.name = 'sysadmin' AND ServerLogin.name LIKE 'L%'
13 ORDER BY [LOGIN Name];
14 GO
```

Role ID	Server Role	LOGIN ID	LOGIN Name	Type
3	sysadmin	262	L1	SQL_LOGIN
3	sysadmin	265	L4	SQL_LOGIN
3	sysadmin	266	L5	SQL_LOGIN
3	sysadmin	267	L6	SQL_LOGIN

- User thuộc các role mới tạo (r1 - r3):

```

1 SELECT
2     RoleP.principal_id AS [Role ID],
3     RoleP.name AS [Role Name],
4     RoleMem.member_principal_id as [User ID],
5     UserP.name AS [User Name],
6     UserP.type_desc AS [Member Type]
7 FROM sys.database_role_members AS RoleMem
8 INNER JOIN sys.database_principals AS RoleP
9     ON RoleMem.role_principal_id = RoleP.principal_id
10 INNER JOIN sys.database_principals AS UserP
11     ON RoleMem.member_principal_id = UserP.principal_id
12 WHERE UserP.name LIKE 'U%' AND RoleP.name LIKE 'r%'
13 ORDER BY [Role Name];
14 GO

```

Role ID	Role Name	User ID	User Name	Member Type
13	r1	7	U1	SQL_USER
14	r2	8	U2	SQL_USER
14	r2	9	U3	SQL_USER
15	r3	10	U4	SQL_USER
15	r3	11	U5	SQL_USER
15	r3	12	U6	SQL_USER

- User thuộc db\_accessadmin

```

1 SELECT
2     RoleP.principal_id AS [Role ID],
3     RoleP.name AS [Role Name],
4     RoleMem.member_principal_id as [User ID],
5     UserP.name AS [User Name],
6     UserP.type_desc AS [Member Type]
7 FROM sys.database_role_members AS RoleMem
8 INNER JOIN sys.database_principals AS RoleP
9     ON RoleMem.role_principal_id = RoleP.principal_id
10 INNER JOIN sys.database_principals AS UserP
11     ON RoleMem.member_principal_id = UserP.principal_id
12 WHERE UserP.name LIKE 'U%' AND RoleP.name LIKE 'db_accessadmin'
13 ORDER BY [Role Name];
14 GO

```

Role ID	Role Name	User ID	User Name	Member Type
16385	db_accessadmin	8	U2	SQL_USER
16385	db_accessadmin	9	U3	SQL_USER



16385	db_accessadmin	10	U4	SQL_USER
16385	db_accessadmin	11	U5	SQL_USER
16385	db_accessadmin	12	U6	SQL_USER

- User thuộc db\_owner

```

1 SELECT
2     RoleP.principal_id AS [Role ID],
3     RoleP.name AS [Role Name],
4     RoleMem.member_principal_id as [User ID],
5     UserP.name AS [User Name],
6     UserP.type_desc AS [Member Type]
7 FROM sys.database_role_members AS RoleMem
8 INNER JOIN sys.database_principals AS RoleP
9     ON RoleMem.role_principal_id = RoleP.principal_id
10 INNER JOIN sys.database_principals AS UserP
11     ON RoleMem.member_principal_id = UserP.principal_id
12 WHERE UserP.name LIKE 'U%' AND RoleP.name LIKE 'db_owner'
13 ORDER BY [Role Name];
14 GO

```

Role ID	Role Name	User ID	User Name	Member Type
16384	db_owner	8	U2	SQL_USER
16384	db_owner	9	U3	SQL_USER
16384	db_owner	10	U4	SQL_USER
16384	db_owner	11	U5	SQL_USER
16384	db_owner	12	U6	SQL_USER

## 2.3. C. PHÂN QUYỀN NGƯỜI DÙNG

**TODO:** Bài 2.C: SQL Script

### 2.3.1. Mô Tả Yêu Cầu

Thực hiện các phát biểu `GRANT` / `DENY` / `REVOKE` trên CSDL **Quản lý đề tài** gồm các bảng sau (Nhóm 2):

- T1: `DETAI`
- T2: `HOIDONG`
- T3: `GV_HDDT`

Tạo các user `U1`, `U2`, `U3` và phân quyền theo yêu cầu.

1. U1 có quyền select, delete trên T1, T3
2. U2 có quyền update, delete trên T2
3. U3 có quyền insert trên T1, T2, T3
4. U1 bị từ chối quyền insert trên T1, T2
5. U2 bị từ chối quyền delete trên T3
6. Thu hồi các quyền của U1 trên T1
7. Thu hồi các quyền của U3 trên T2

### 2.3.2. Tạo User - Chuẩn Bị Môi Trường

```
1 CREATE USER U1 WITHOUT LOGIN;
2 CREATE USER U2 WITHOUT LOGIN;
3 CREATE USER U3 WITHOUT LOGIN;
```

- Kiểm tra kết quả:

```
1 SELECT name, type_desc FROM sys.database_principals
2 WHERE name IN ('U1', 'U2', 'U3');
```

name	type_desc
U1	SQL_USER
U2	SQL_USER
U3	SQL_USER

### 2.3.3. GRANT - U1 có quyền SELECT, DELETE trên T1, T3

#### 2.3.3.1. GRANT

```
1 GRANT SELECT, DELETE ON DETAI TO U1;
2 GRANT SELECT, DELETE ON GV_HDDT TO U1;
```

#### 2.3.3.2. Kiểm tra

```
1 EXECUTE AS USER = 'U1';
2 SELECT * FROM DETAI;
3 REVERT;
```

MSDT	TENDT
97001	Quản lý thư viện
97002	Nhận dạng vân tay
97003	Bán đấu giá trên mạng
97004	Quản lý siêu thị

97005	Xử lý ảnh
97006	Hệ giải toán thông minh

### 2.3.4. GRANT - U2 có quyền UPDATE, DELETE trên T2

#### 2.3.4.1. GRANT

```
1 GRANT UPDATE, DELETE ON HOIDONG TO U2;
```

#### 2.3.4.2. Kiểm tra

- DELETE

```
1 EXECUTE AS USER = 'U2';
2 DELETE FROM HOIDONG WHERE 1 = '0';
3 REVERT;
4 GO
```

```
1 Commands completed successfully.
```

- UPDATE

```
1 EXECUTE AS USER = 'U2';
2 UPDATE HOIDONG SET PHONG = 6969;
3 REVERT;
```

MSHD	PHONG	TINHTRANG	MSGV
1	6969	Thật	201
2	6969	Thật	202
3	6969	Thật	203

### 2.3.5. GRANT - U3 có quyền INSERT trên T1, T2, T3

#### 2.3.5.1. GRANT

```
1 GRANT INSERT ON DETAI TO U3;
2 GRANT INSERT ON HOIDONG TO U3;
3 GRANT INSERT ON GV_HDDT TO U3;
```

#### 2.3.5.2. Kiểm tra

- INSERT trên HOIDONG

```

1 EXECUTE AS USER = 'U3';
2 INSERT INTO HOIDONG (MSHD, PHONG, TGBD, NGÀYHD, TINHTRANG, MSGV) VALUES
3 (4, 2, '07:00', '2014-11-29', N'Thật', 201);
4 REVERT;

```

MSHD	PHONG	TINHTRANG	MSGV
1	6969	Thật	201
2	6969	Thật	202
3	6969	Thật	203
4	2	Thật	201

### 2.3.6. DENY - U1 bị từ chối quyền INSERT trên T1, T2

#### 2.3.6.1. DENY

```

1 DENY INSERT ON DETAI TO U1;
2 DENY INSERT ON HOIDONG TO U1;

```

#### 2.3.6.2. Kiểm tra

```

1 EXECUTE AS USER = 'U1';
2 INSERT INTO DETAI (MSDT, TENDT) VALUES
3 ('970553', N'Quản lý thư viện2');
4 REVERT;

```

```

1 Msg 229, Level 14, State 5, Line 9
2 The INSERT permission was denied on the object 'DETAI', database 'IE103-BTTH2', schema
'dbo'.

```

### 2.3.7. DENY - U2 bị từ chối quyền DELETE trên T3

#### 2.3.7.1. DENY

```

1 DENY DELETE ON GV_HDDT TO U2;

```

#### 2.3.7.2. Kiểm tra

```

1 EXECUTE AS USER = 'U2';
2 DELETE FROM GV_HDDT WHERE 1 = 0;
3 REVERT;

```

```

1 The DELETE permission was denied on the object 'GV_HDDT', database 'IE103-BTTH2', schema
'dbo'.

```

## 2.3.8. REVOKE - Thu hồi các quyền của U1 trên T1

### 2.3.8.1. REVOKE

```
1 REVOKE SELECT, DELETE ON DETAI FROM U1;
```

### 2.3.8.2. Kiểm tra

```
1 EXECUTE AS USER = 'U1';  
2 SELECT * FROM DETAI;  
3 REVERT;
```

```
1 The SELECT permission was denied on the object 'DETAI', database 'IE103-BTTH2', schema 'dbo'.
```

## 2.3.9. REVOKE - Thu hồi các quyền của U3 trên T2

### 2.3.9.1. REVOKE

```
1 REVOKE INSERT ON HOIDONG FROM U3;
```

### 2.3.9.2. Kiểm tra

```
1 EXECUTE AS USER = 'U3';  
2 INSERT INTO HOIDONG (MSHD, PHONG, TGBD, NGAYHD, TINHTRANG, MSGV) VALUES  
3 (5, 3, '08:00', '2014-11-30', N'Thật', 202);  
4 REVERT;
```

```
1 The INSERT permission was denied on the object 'HOIDONG', database 'IE103-BTTH2', schema 'dbo'.
```