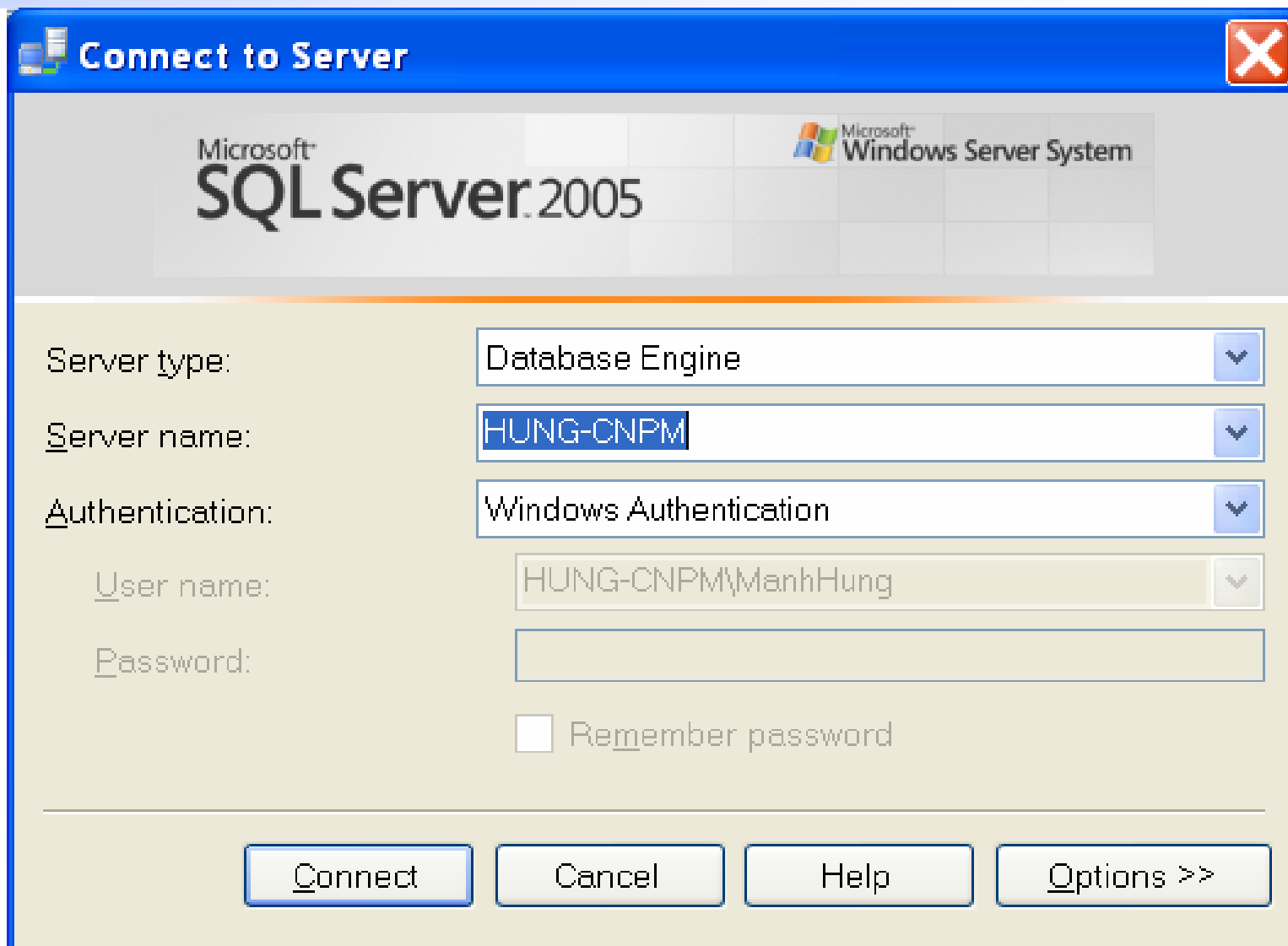


Chương 2: MS SQL Server Management Studio (MSt) và T-SQL

- Khởi động MSt và kết nối đến SQL Server
- Sử dụng các Server đã đăng ký
- Object Explorer
- Xem thông tin trên bảng Summary
- Truy vấn dữ liệu
- Thiết lập môi trường MSt (Thiết lập tùy chọn Startup, Hiển thị kết quả, Phím tắt,...)
- Cú pháp của T-SQL (Identifiers, Variables, Data types, Flow-Control Statements, try/catch,...)
- Biến hệ thống @@Identity
- Tham số môi trường Quoted_Identifier

Khởi động MS



Connect to Server

Microsoft SQL Server 2005

Microsoft Windows Server System

Server type: Database Engine

Server name: HUNG-CNPM

Authentication: Windows Authentication

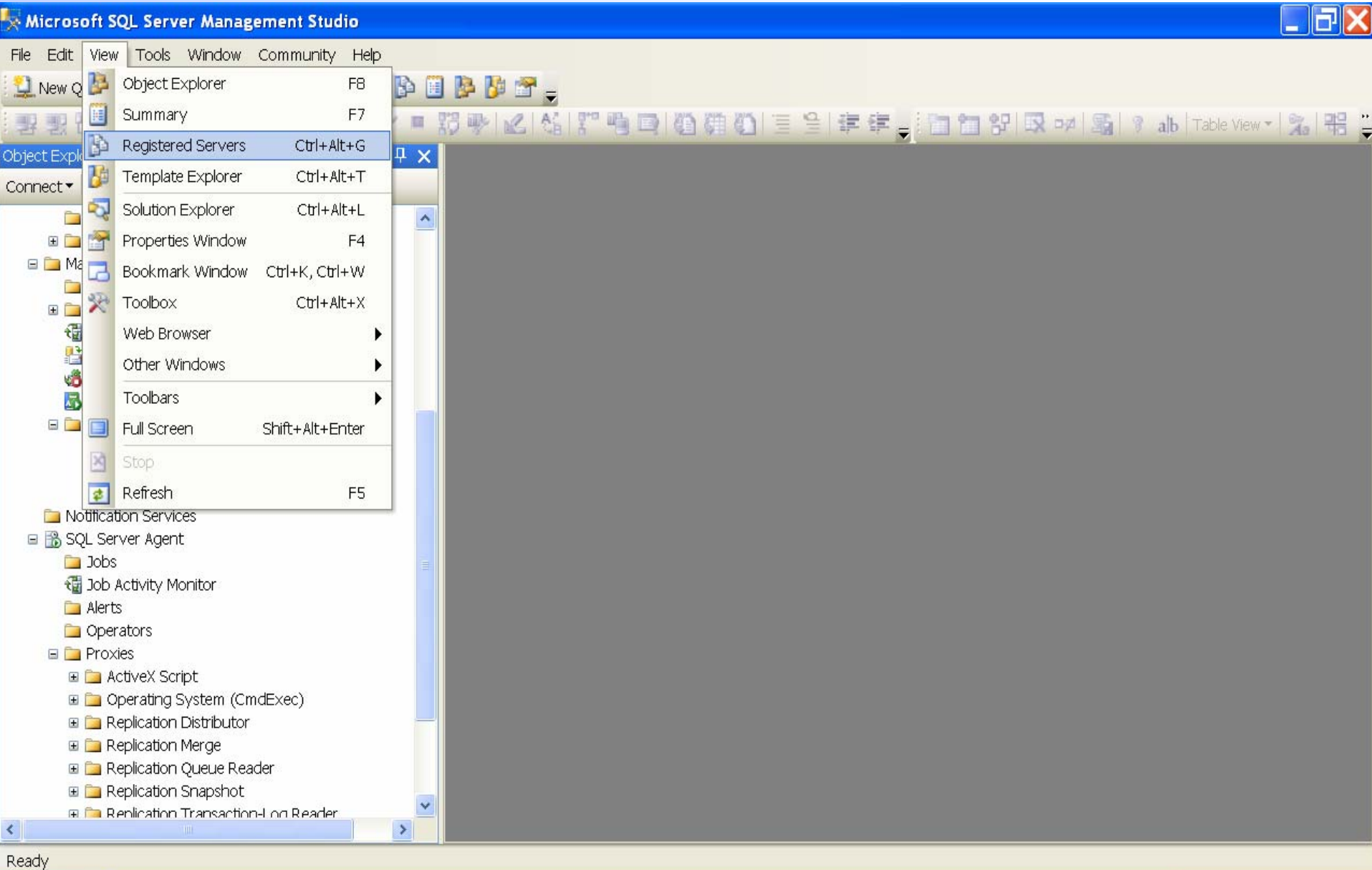
User name: HUNG-CNPM\ManhHung

Password:

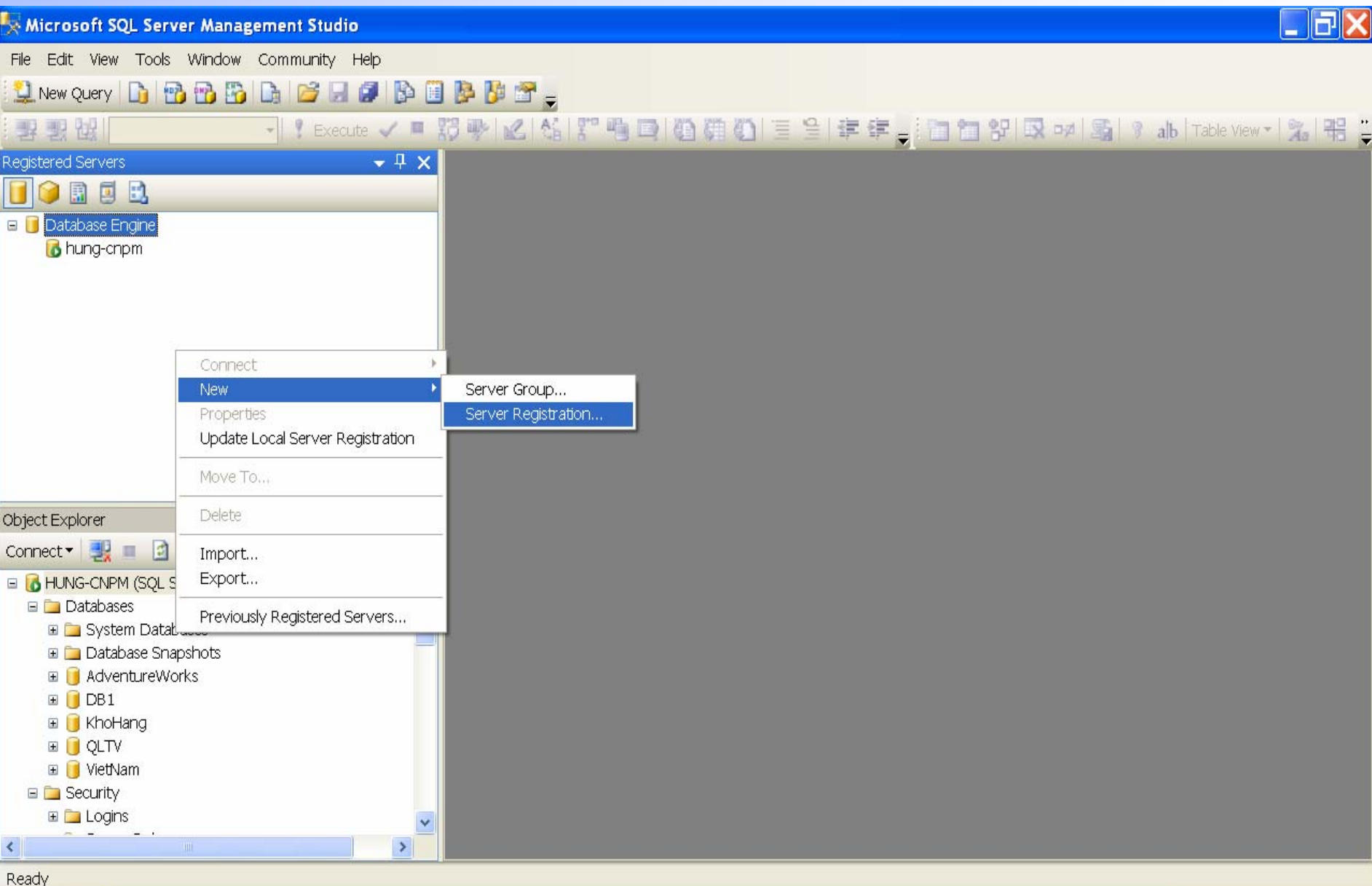
☐ Remember password

Connect Cancel Help Options >>

Sử dụng các Server đã đăng ký



Tạo kết nối mới

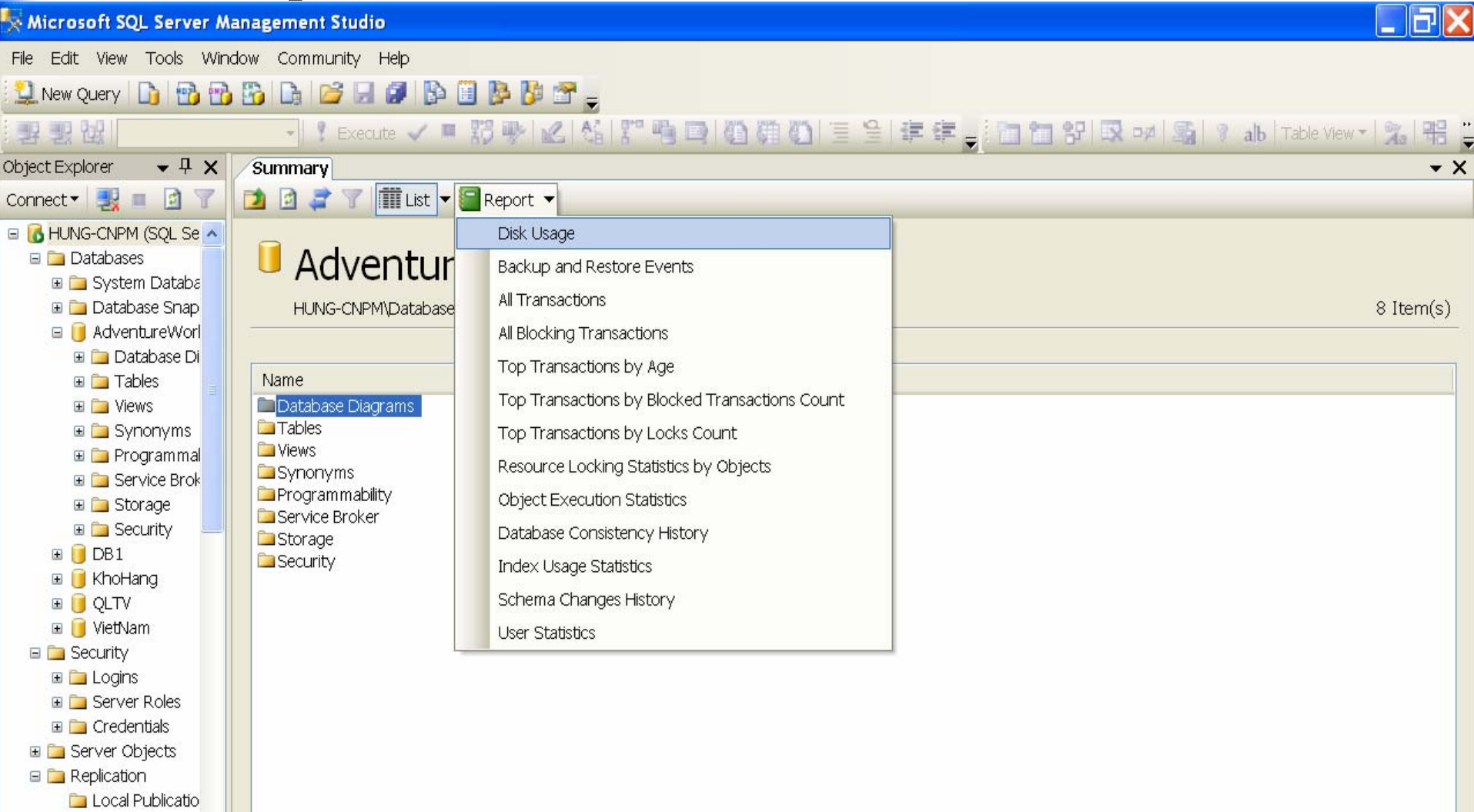


Object Explorer

- Xem và thay đổi thuộc tính CSDL: kích phải chuột lên một CSDL để xem các thông tin về: recovery model, tự động nén, tự động tăng kích cỡ CSDL, tên và vị trí các file CSDL,...
- Bảo mật:
 - Kích phải chuột vào folder Security để tạo mới một login.
 - Kích phải chuột vào folder Security trong CSDL để tạo mới một user trong CSDL.

Xem thông tin bảng Summary

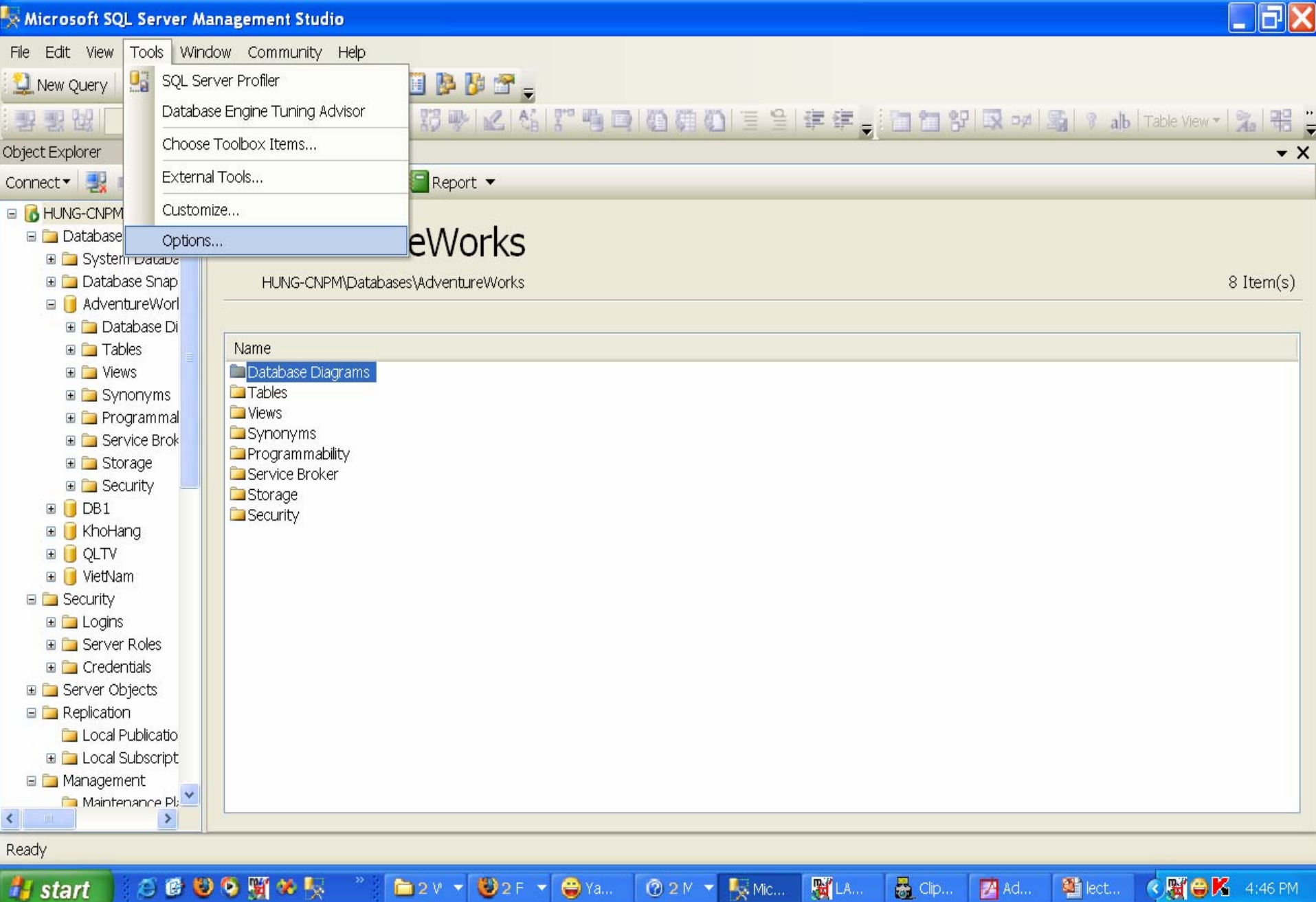
- Ấn F7 để xem thông tin Summary.
- Kích đúp vào CSDL AdventureWorks



Truy vấn dữ liệu sử dụng T-SQL

- Chọn Database Engine Query
- Nhập câu lệnh T-SQL
- Ấn F5 để thực hiện

Thiết lập môi trường MS



Cú Pháp Của T-SQL

Identifiers – Các định danh

Cách đặt tên table, view, stored procedure, index, tên biến,...

- có thể dài từ 1 đến 128 ký tự theo mã Unicode (gồm chữ cái, chữ số, ký hiệu)
- ký tự đầu tiên phải là: chữ cái hoặc @, #, _, or \$.
- có thể sử dụng khoảng trắng trong các định danh, tuy nhiên để truy cập các định danh này bạn phải đặt nó trong dấu nháy kép hoặc ngoặc vuông [].

VD: 7007Customers; @Variable; #tblTMP; [Ho va Ten]

Database Object Qualifiers – Định danh đầy đủ của một đối tượng CSDL:

[[[server.][database.][schema].]database_object

Kiểu dữ liệu

Kiểu dữ liệu	Diễn giải
Char(n), varchar(n max)	String – xâu ký tự có kích thước tối đa 8000 ký tự. Varchar(max) cho phép kích thước lưu trữ đến 2GB
Binary(n)	Binary – lưu dữ liệu nhị phân
Bit	Lưu giá trị 0 hoặc 1
Int, Smallint,bigint,tinyint	Integer
Float – 8b,Real -4b	Lưu dữ liệu số thực
Decimal[(s[,p])], numeric[(s[,p])]	Lưu số chính xác. S: 1-38;P: 0-S
Datetime, smalldatetime	Lưu trữ ngày giờ
Money – 8 bytes, Smallmoney – 4 byte	Lưu trữ các giá trị tiền tệ. Các hằng số tiền có thể có tiền tố \$.
Nchar(n), Nvarchar(n)	Lưu trữ văn bản theo mã Unicode có kích thước tối đa là 4000 ký tự. Nvarchar(max) cho phép kích thước lưu trữ đến 2GB
Text, Image, Ntext	Lưu dữ liệu nhị phân, văn bản dài không quá $2^{31} - 1$ (2,147,483,647) ký tự

Kiểu dữ liệu TimesTamp - 4 byte

- Trường kiểu **TimesTamp** được tự động cập nhật mỗi khi bản ghi đó bị thay đổi.

VD: Ta có bảng nhân viên có rất nhiều thông tin (hoten, ngaysinh, diachi, capbac, chucvu, luong, ..., TM **TimesTamp**). Các thông tin về cấp bậc, chức vụ, lương có thể thay đổi liên tục.

Xác định nhân viên có sự thay đổi gần đây nhất:

```
Select * from NhanVien where TM in (select max(TM) from NhanVien)
```

Uniqueidentifier – 16 bytes

Giá trị mà uniqueidentifier lưu trữ còn được gọi là globally unique identifiers (GUIDs). Giá trị GUIDs đảm bảo là duy nhất trên toàn thế giới.

GUIDs are generated using the identification number of the network card and a unique number obtained from the computer's clock.

```
CREATE TABLE MyUniqueTable
```

```
  (UniqueColumn UNIQUEIDENTIFIER    DEFAULT NEWID(),  
   Characters    VARCHAR(10) )
```

```
GO
```

```
INSERT INTO MyUniqueTable(Character) VALUES ('abc')
```

```
INSERT INTO MyUniqueTable VALUES (NEWID(), 'def')
```

```
GO
```

Kiểu dữ liệu XML

```
Create table dbo.TableXML(  
    Id int,  
    CompList xml)
```

```
go
```

```
INSERT INTO dbo.TableXML(Id, CompList)  
VALUES(123,'<CompList><CZ101/><AZ401/><BZ407/><  
    /CompList>')
```

Biến trong SQL Server 2005

Khai báo:

Cú pháp: Declare @tenbien <kieudulieu>[,@tenbien <kieudulieu> ...]

VD: declare @dem int , @tong int

Gán giá trị cho biến:

Dùng lệnh SET:

Set @dem=10

Set @tong=@tong+1

Dùng lệnh Select:

Select @dem=10

Select @tong=@tong+1

In giá trị của biến:

Select @dem

Print @dem

Biến trong SQL Server 2005 (2)

Gán giá trị cho biến trong lệnh Update

use asset5

go

declare @mnsCost smallmoney

Update Inventory

Set @mnsCost = Cost = Cost *4

Where InventoryId = 6

Các lệnh cơ bản của T-SQL

- **Chú thích**
 - Chú thích dòng đơn: -- chú thích
 - Chú thích nhiều dòng: /* nhiều dòng chú thích */
- **Góm nhóm các câu lệnh thành một khối**
BEGIN
{sql_statement|statement_block}
END
- **Lệnh nhảy GOTO**
Label:
GOTO *label*

Lệnh rẽ nhánh

Cú pháp: IF *Boolean_expression*

{sql_statement/statement_block}

[ELSE

{sql_statement/statement_block}]

```
declare @i float
```

```
select @i=RAND() --RAND() tra ve gia tri kieu float 0-1
```

```
print 'gia tri nhan duoc ' + cast(@i as varchar(10))
```

```
if (@i>0.5)
```

```
--Begin
```

```
print 'Gia tri lon hon 0.5'
```

```
print '!!!'
```

```
--End else
```

```
print 'Gia tri nho hon 0.5'
```

Ví dụ: Kết hợp GOTO và IF

```
declare @i float
```

```
Lap:
```

```
select @i=RAND()
```

```
--RAND() tra ve gia tri kieu float 0-1
```

```
print 'Gia tri nhan duoc ' + cast(@i as varchar(10))
```

```
if (@i>0.5)
```

```
Begin
```

```
print 'Gia tri lon hon 0.5'
```

```
print '!!!'
```

```
goto lap
```

```
End
```

```
Else print 'Gia tri nho hon 0.5'
```

Vòng lặp While

```
WHILE Boolean_condition  
    {sql_statement | statement_block}  
    [ BREAK ]  
    {sql_statement | statement_block}  
    [ CONTINUE ]
```

```
DECLARE @counter smallint
```

```
SET @counter = 1
```

```
WHILE @counter < 5
```

```
    BEGIN
```

```
        SELECT @counter [So thu tu], RAND(@counter)  
        Random_Number
```

```
        SET @counter = @counter + 1
```

```
    END
```

Break - Thoát khỏi vòng WHILE

```
DECLARE @counter smallint
```

```
SET @counter = 1
```

```
WHILE @counter < 10
```

```
    BEGIN
```

```
        SELECT @counter [Số thu từ], RAND(@counter)  
        Random_Number
```

```
        SET @counter = @counter + 1
```

```
        if @counter >= 5
```

```
            break
```

```
    END
```

CONTINUE -Tiếp tục vòng WHILE

```
DECLARE @counter smallint
```

```
DECLARE @c2 smallint
```

```
SET @counter = 1
```

```
set @c2=1
```

```
WHILE @counter < 10
```

```
    BEGIN
```

```
        SELECT @c2 [Số thu tu], RAND(@counter) Random_Number
```

```
        set @c2=@c2+1
```

```
        if @c2>50
```

```
            break
```

```
        if @counter>=5
```

```
            continue
```

```
        SET @counter = @counter + 1
```

```
    END
```

Hàm CASE

Cú pháp 1:

CASE <input_expression>

 WHEN when_expression THEN result_expression

 [WHEN ...]

 [ELSE else_result_expression]

END

Cú pháp 2:

CASE

 WHEN *Boolean_expression* THEN *result_expression*

 [WHEN ...]

 [*ELSE else_result_expression*]

END

Hàm CASE (2)

```
declare @st varchar(100)
```

```
declare @i float
```

```
set @i=RAND()
```

```
SELECT  @st =
```

```
        CASE
```

```
            WHEN @i<0.2 THEN 'Gia tri nho hon 0.2'
```

```
            WHEN @i<0.4 THEN 'Gia tri nho hon 0.4'
```

```
        ELSE
```

```
            'Cac gia tri khac'
```

```
        END
```

```
print @st
```

Hàm CASE (3)

```
USE AdventureWorks;
GO
SELECT ProductNumber, Category =
    CASE ProductLine
        WHEN 'R' THEN 'Road'
        WHEN 'M' THEN 'Mountain'
        WHEN 'T' THEN 'Touring'
        WHEN 'S' THEN 'Other sale items'
        ELSE 'Not for sale'
    END, Name FROM Production.Product
ORDER BY ProductNumber;GO
```

Hàm CASE (4)

```
USE AdventureWorks;GO
SELECT ProductNumber, Name, 'Price Range' =
CASE
    WHEN ListPrice = 0 THEN 'Mfg item - not for resale'
    WHEN ListPrice < 50 THEN 'Under $50'
    WHEN ListPrice >= 50 and ListPrice < 250 THEN 'Under $250'
    WHEN ListPrice >= 250 and ListPrice < 1000 THEN 'Under
$1000'
    ELSE 'Over $1000'
END
FROM Production.Product
ORDER BY ProductNumber ;
GO
```

WAITFOR {DELAY '*time*'|TIME '*time*'}

DELAY '*time*' – khoảng thời gian mà SQL Server phải chờ.

TIME '*time*' – yêu cầu SQL Server chờ đến thời điểm chỉ định.

Ví dụ:

```
WAITFOR Time '12:47:00'
```

Print ‘Câu lệnh này chỉ được chạy sau 12:47:00’

```
WAITFOR Delay '00:00:01'
```

WAITFOR {DELAY '*time*'|TIME '*time*'}

Use tempdb go

```
alter PROCEDURE time_delay @@DELAYLENGTH char(8)
```

```
AS
```

```
DECLARE @@RETURNINFO varchar(255)
```

```
BEGIN
```

```
    WAITFOR DELAY @@DELAYLENGTH
```

```
    SELECT @@RETURNINFO = 'A total time of ' +
```

```
        SUBSTRING(@@DELAYLENGTH, 1, 2) +  
        ' hours, ' +
```

```
        SUBSTRING(@@DELAYLENGTH, 4, 2) +  
        ' minutes, and ' +
```

```
        SUBSTRING(@@DELAYLENGTH, 7, 2) +  
        ' seconds, ' +
```

```
        'has elapsed! Your time is up.'
```

```
    PRINT @@RETURNINFO
```

```
END GO
```

```
EXEC time_delay '00:00:10'
```

Try/Catch

Cú pháp:

BEGIN TRY

{ *sql_statement* | *statement_block* }

END TRY

BEGIN CATCH

{ *sql_statement* | *statement_block* }

END CATCH [;]

try/catch phải trong một batch

Ví dụ: Đoạn code sau sẽ phát sinh lỗi

```
BEGIN TRY
```

```
    SELECT *          FROM sys.messages
```

```
    WHERE message_id = 21;
```

```
END TRY
```

```
GO
```

```
BEGIN CATCH
```

```
    SELECT ERROR_NUMBER() AS ErrorNumber;
```

```
END CATCH;
```

```
GO
```


Các Error Functions

`ERROR_NUMBER()` returns the error number.

`ERROR_MESSAGE()` returns the complete text of the error message.
The text includes the values supplied for any substitutable parameters such as lengths, object names, or times.

`ERROR_SEVERITY()` returns the error severity. (Trả về sự nghiêm trọng của lỗi)

`ERROR_STATE()` returns the error state number.

`ERROR_LINE()` returns the line number inside the routine that caused the error.

`ERROR_PROCEDURE()` returns the name of the stored procedure or trigger where the error occurred.

Severity Level

Severity	Description
0-9	Informational messages that return status information or report errors that are not severe. The Database Engine does not raise system errors with severities of 0 through 9.
10	Informational messages that return status information or report errors that are not severe. For compatibility reasons, the Database Engine converts severity 10 to severity 0 before returning the error information to the calling application.
11-16	Indicate errors that can be corrected by the user.
11	Indicates that the given object or entity does not exist.
12	A special severity for queries that do not use locking because of special query hints. In some cases, read operations performed by these statements could result in inconsistent data, since locks are not taken to guarantee consistency.
13	Indicates transaction deadlock errors.
14	Indicates security-related errors, such as permission denied.
15	Indicates syntax errors in the Transact-SQL command.
16	Indicates general errors that can be corrected by the user.
17-19	Indicate software errors that cannot be corrected by the user. Inform your system administrator of the problem.
17	Indicates that the statement caused SQL Server to run out of resources (such as memory, locks, or disk space for the database) or to exceed some limit set by the system administrator.
18	Indicates a problem in the Database Engine software, but the statement completes execution, and the connection to the instance of the Database Engine is maintained. The system administrator should be informed every time a message with a severity level of 18 occurs.

Severity	Description
19	Indicates that a nonconfigurable Database Engine limit has been exceeded and the current batch process has been terminated. Error messages with a severity level of 19 or higher stop the execution of the current batch. Severity level 19 errors are rare and must be corrected by the system administrator or your primary support provider. Contact your system administrator when a message with a severity level 19 is raised. Error messages with a severity level from 19 through 25 are written to the error log.
20-25	Indicate system problems and are fatal errors, which means that the Database Engine task that is executing a statement or batch is no longer running. The task records information about what occurred and then terminates. In most cases, the application connection to the instance of the Database Engine may also terminate. If this happens, depending on the problem, the application might not be able to reconnect. Error messages in this range can affect all of the processes accessing data in the same database and may indicate that a database or object is damaged. Error messages with a severity level from 19 through 25 are written to the error log.
20	Indicates that a statement has encountered a problem. Because the problem has affected only the current task, it is unlikely that the database itself has been damaged.
21	Indicates that a problem has been encountered that affects all tasks in the current database, but it is unlikely that the database itself has been damaged.
22	Indicates that the table or index specified in the message has been damaged by a software or hardware problem. Severity level 22 errors occur rarely. If one occurs, run DBCC CHECKDB to determine whether other objects in the database are also damaged. The problem might be in the buffer cache only and not on the disk itself. If so, restarting the instance of the Database Engine corrects the problem. To continue working, you must reconnect to the instance of the Database Engine; otherwise, use DBCC to repair the problem. In some cases, you may have to restore the database. If restarting the instance of the Database Engine does not correct the problem, then the problem is on the disk. Sometimes destroying the object specified in the error message can solve the problem. For example, if the message reports that the instance of the Database Engine has found a row with a length of 0 in a nonclustered index, delete the index and rebuild it.
23	Indicates that the integrity of the entire database is in question because of a hardware or software problem. Severity level 23 errors occur rarely. If one occurs, run DBCC CHECKDB to determine the extent of the damage. The problem might be in the cache only and not on the disk itself. If so, restarting the instance of the Database Engine corrects the problem. To continue working, you must reconnect to the instance of the Database Engine; otherwise, use DBCC to repair the problem. In some cases, you may have to restore the database.
24	Indicates a media failure. The system administrator may have to restore the database. You may also have to call your hardware vendor.

Hàm phát sinh lỗi - RAISERROR

RAISERROR ({ *msg_id* | *msg_str* } { , *severity* , *state* }
[, *argument* [,...*n*]])
[WITH *option* [,...*n*]]

- *msg_id* là id của thông báo lỗi trong SQL Server. Các thông báo này có trong sys.messages, có thể xem bằng `select * from sys.messages`
- *msg_str*: thông báo lỗi do người dùng định nghĩa
- *severity*: mức độ của lỗi (do hệ thống hay do người sử dụng).
- *State*: trạng thái của lỗi.
- Ví dụ:
 - RAISERROR ('Thông báo lỗi', 16, 1)

Error messages

- Các thông báo lỗi được định nghĩa trước trong View: sys.messages của CSDL master.
 - Xem các thông báo lỗi: `select * from sys.messages where message_id > 13000`
 - Phát sinh thông báo lỗi theo message_id đã định nghĩa trước:
 - `select * from sys.messages where message_id=14148`
 - `raiserror (14148 ,16,1,'Gia tri khong dung')`

Error messages-Ghi các error message ra log (2)

- Cho phép ghi các error message ra log:
 - `sp_altermessage [@message_id =] message_number , [@parameter =] 'write_to_log' , [@parameter_value =] 'value'`
 - `'write_to_log'`: must be set to WITH_LOG or NULL.
 - `@parameter_value` is **true** or **false**
 - Ví dụ: `sp_altermessage 55001, 'WITH_LOG', 'true';`

Error messages - Thêm thông báo lỗi (3)

```
sp_addmessage [ @msgnum = ] msg_id ,  
    [ @severity = ] severity , [ @msgtext = ] 'msg'  
    [ , [ @lang = ] 'language' ] [ , [ @with_log = ] 'with_log' ] [ ,  
    [ @replace = ] 'replace' ]
```

Ví dụ:-- Các thông báo lỗi do NSD định nghĩa phải có ID >50000

```
EXEC sp_addmessage 50001, 16, 'Thong bao loi do Boby dinh nghĩa'  
raiserror (50001,16,1)  
EXEC sp_addmessage 50002, 16, "Thong bao loi '%s' do CIA dinh  
    nghĩa"  
raiserror (50002,16,1, 've vu no phot pho')  
EXEC sp_addmessage 50003, 16, 'Thong bao loi co ghi ra  
    log','us_english','true'  
raiserror (50003,16,1)  
- Để sửa một thông báo cũ (message_id>50000) ta dùng tham số  
    replace
```


usp_GetErrorInfo

```
USE AdventureWorks;
IF OBJECT_ID ('usp_GetErrorInfo', 'P') IS NOT NULL
    DROP PROCEDURE usp_GetErrorInfo;
CREATE PROCEDURE usp_GetErrorInfo
AS
    SELECT  ERROR_NUMBER() AS ErrorNumber,
            ERROR_SEVERITY() AS ErrorSeverity, ERROR_STATE() as
            ErrorState, ERROR_PROCEDURE() as ErrorProcedure,
            ERROR_LINE() as ErrorLine, ERROR_MESSAGE() as
            ErrorMessage;
GO --Thử nghiệm việc phát sinh lỗi chia cho không
BEGIN TRY    -- Generate divide-by-zero error.
    SELECT 1/0;
END TRY
BEGIN CATCH -- Execute the error retrieval routine.
    EXECUTE usp_GetErrorInfo;
END CATCH;
```

Try/catch và transaction

```
USE AdventureWorks;
```

```
IF OBJECT_ID (N'my_books', N'U') IS NOT NULL
```

```
DROP TABLE my_books;
```

```
CREATE TABLE my_books (Isbn int PRIMARY KEY, Title  
    NVARCHAR(100) );
```

Try/catch và transaction(2)

```
BEGIN TRY
    BEGIN TRANSACTION;
    ALTER TABLE my_books DROP COLUMN author;-- truong nay ko co
    COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    SELECT      ERROR_NUMBER() as ErrorNumber,
    ERROR_MESSAGE() as ErrorMessage;
    IF (XACT_STATE()) = -1 -- Test if the transaction is uncommittable.
    BEGIN PRINT N'The transaction is in an uncommittable state. ' +
        'Rolling back transaction.'
        ROLLBACK TRANSACTION;
    END;
    IF (XACT_STATE()) = 1 -- Test if the transaction is active and valid.
    BEGIN PRINT N'The transaction is committable.Committing transaction.'
        COMMIT TRANSACTION;
    END;
END CATCH;
```

Sử dụng try/catch để bắt deadlock

```
USE AdventureWorks;
CREATE TABLE my_sales ( Itemid INT PRIMARY KEY,Sales INT not null );

INSERT my_sales (itemid, sales) VALUES (1, 1);
INSERT my_sales (itemid, sales) VALUES (2, 1);

CREATE PROCEDURE usp_MyErrorLog
AS
    PRINT
        'Error ' + CONVERT(VARCHAR(50), ERROR_NUMBER()) +
        ', Severity ' + CONVERT(VARCHAR(5), ERROR_SEVERITY()) +
        ', State ' + CONVERT(VARCHAR(5), ERROR_STATE()) +
        ', Line ' + CONVERT(VARCHAR(5), ERROR_LINE());
    PRINT      ERROR_MESSAGE();
GO
```

Sử dụng try/catch để bắt deadlock(2)

```
USE AdventureWorks;
DECLARE @retry INT;
SET @retry = 5;
WHILE (@retry > 0)
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;
        UPDATE my_sales      SET sales = sales + 1
        WHERE itemid = 1;
        WAITFOR DELAY '00:00:13';
        UPDATE my_sales      SET sales = sales + 1
        WHERE itemid = 2;

        SET @retry = 0;
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH -- phát sinh lỗi do không thể đồng
        thời update
        IF (ERROR_NUMBER() = 1205)--lỗi 1205
            deadlock
            SET @retry = @retry - 1;
        ELSE
            SET @retry = -1;
        -- print error information
        EXECUTE usp_MyErrorLog;
        IF XACT_STATE() <> 0
            ROLLBACK TRANSACTION;
    END CATCH;
END; -- end while loop
```

```
USE AdventureWorks;
DECLARE @retry INT;
SET @retry = 5;
WHILE (@retry > 0)
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;
        UPDATE my_sales      SET sales = sales + 1
        WHERE itemid = 2;
        WAITFOR DELAY '00:00:07';
        UPDATE my_sales      SET sales = sales + 1
        WHERE itemid = 1;
        SET @retry = 0;
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF (ERROR_NUMBER() = 1205)
            SET @retry = @retry - 1;
        ELSE
            SET @retry = -1;
        -- print error information
        EXECUTE usp_MyErrorLog;
        IF XACT_STATE() <> 0
            ROLLBACK TRANSACTION;
    END CATCH;
END; -- end while loop
```

Try/catch và Raiserror

```
USE AdventureWorks; -- Return if there is no error information to retrieve.
CREATE PROCEDURE usp_Re_throwError AS
    IF ERROR_NUMBER() IS NULL
        RETURN;
    DECLARE @ErrorMessage NVARCHAR(4000), @ErrorNumber
        INT, @ErrorSeverity INT, @ErrorState INT, @ErrorLine INT, @ErrorProcedure
        NVARCHAR(200);
    SELECT --Gan cac bien
        @ErrorNumber = ERROR_NUMBER(), @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE(), @ErrorLine = ERROR_LINE(),
        @ErrorProcedure = ISNULL(ERROR_PROCEDURE(), '-'); -- error information.
    SELECT @ErrorMessage = N'Error %d, Level %d, State %d, Procedure %s, Line %d, '
        + 'Message: '+ ERROR_MESSAGE();
    RAISERROR
        ( @ErrorMessage, @ErrorSeverity, 1, @ErrorNumber, -- parameter: original error
        number.
        @ErrorSeverity, -- parameter: original error severity.
        @ErrorState, -- parameter: original error state.
        @ErrorProcedure, -- parameter: original error procedure name.
        @ErrorLine -- parameter: original error line number.
        );
GO
```

Try/catch và Raiserror(2)

```
CREATE PROCEDURE usp_GenerateError
AS
    BEGIN TRY    -- statement will generate a constraint violation error.
        DELETE FROM Production.Product WHERE ProductID = 980;
    END TRY
    BEGIN CATCH
        EXEC usp_Re_throwError;
    END CATCH;
GO
BEGIN TRY -- outer TRY
    EXECUTE usp_GenerateError; -- Call the procedure to generate an error.
END TRY
BEGIN CATCH -- outer CATCH
    SELECT
        ERROR_NUMBER() as ErrorNumber,
        ERROR_MESSAGE() as ErrorMessage;
END CATCH;
```

Các lỗi mà try/catch không bắt được

Try/catch: không bắt được các lỗi cú pháp, lỗi biện dịch

```
USE AdventureWorks; GO
BEGIN TRY
    PRINT N'Starting execution';
    SELECT * FROM
    NonExistentTable;-- bang nay ko co.
END TRY
BEGIN CATCH
    SELECT
        ERROR_NUMBER() AS
        ErrorNumber,
        ERROR_MESSAGE() AS
        ErrorMessage;
END CATCH;
GO
.
```

Try/catch: cho phép bắt các lỗi mức trên

```
USE AdventureWorks;
IF OBJECT_ID ('usp_MyError', 'P') IS NOT NULL
    DROP PROCEDURE usp_MyError;
CREATE PROCEDURE usp_MyError
AS
SELECT * FROM NonExistentTable;
```

```
BEGIN TRY
EXECUTE usp_MyError;
END TRY
BEGIN CATCH
    SELECT
        ERROR_NUMBER() AS ErrorNumber,
        ERROR_MESSAGE() AS ErrorMessage;
END CATCH;
GO
```


Hàm hệ thống - @@IDENTITY

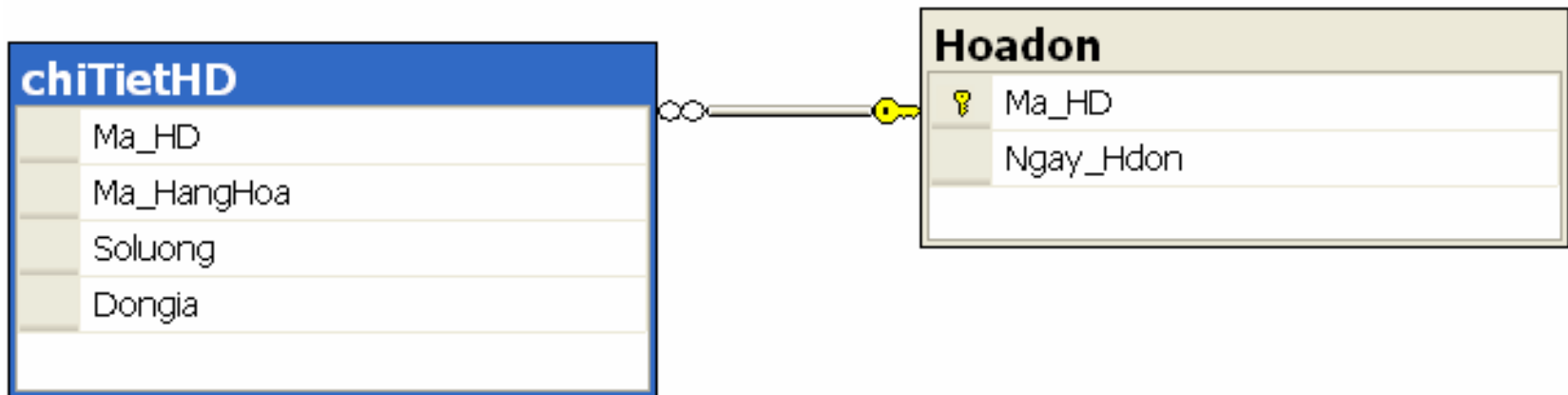
Hàm @@IDENTITY cung cấp giá trị tự động tăng trong bảng

VD:use tempDB go

Create table Hoadon (Ma_HD int IDENTITY primary key, Ngay_Hdon
DateTime) go

Create table chiTietHD

(Ma_HD int Foreign key references HoaDon(Ma_HD), Ma_HangHoa int,
Soluong int, Dongia float) Go



Một số hàm hệ thống

Declare @Ident int

Insert into Hoadon(ngay_Hdon) values('10/10/2006')

Set @Ident=@@Identity

Print 'Ma hoa don hien thoi '+ convert(varchar(4),@Ident)

Insert into chiTietHD values(@Ident,1,100,100.1)

Insert into chiTietHD values(@Ident,2, 200,200.2)

Các thao tác trên Query Analyzer

- Khởi động Query analyzer

Thực hiện	Ctrl+E hoặc F5
Tìm kiếm	Ctrl+F
Biến đổi thành chữ hoa khối được đánh dấu	Ctrl+Shift+U
Biến đổi thành chữ thường khối được đánh dấu	Ctrl+Shift+L
Hiển thị kết quả trong text	Ctrl+T
Hiển thị kết quả trong Grid	Ctrl+D
Trợ giúp	F1
Xem trợ giúp câu lệnh được đánh dấu	Shift+F1

Các tham số - QUOTED_IDENTIFIER

- Tham số **QUOTED_IDENTIFIER** có thể ở một trong 2 trạng thái: on/off

CP:

SET QUOTED_IDENTIFIER Off

SET QUOTED_IDENTIFIER On

VD: SELECT "x" FROM T

Nếu QUOTED_IDENTIFIER is set to ON thì “x” có nghĩa là tên cột x.

Nếu QUOTED_IDENTIFIER is set to OFF, thì "x" có nghĩa là hằng ký tự x.

VD: SET QUOTED_IDENTIFIER On

Use pubs

select "au_id" from authors

SET QUOTED_IDENTIFIER Off

select "au_id" from authors

Các tham số - QUOTED_IDENTIFIER

-- Them cac nhay don, kep vao hang xau ki tu

```
SET QUOTED_IDENTIFIER OFF
```

```
GO
```

```
USE AdventureWorks
```

```
IF EXISTS(SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES  
          WHERE TABLE_NAME = 'Test')
```

```
  DROP TABLE dbo.Test
```

```
GO
```

```
USE AdventureWorks
```

```
CREATE TABLE dbo.Test (ID INT, String VARCHAR(30))
```

```
GO
```

-- Literal strings can be in single or double quotation marks.

```
INSERT INTO dbo.Test VALUES (1, "Text in single quotes")
```

```
INSERT INTO dbo.Test VALUES (2, "Text in single quotes")
```

```
INSERT INTO dbo.Test VALUES (3, 'Text with 2 "" single quotes')
```

```
INSERT INTO dbo.Test VALUES (4, "Text in double quotes")
```

```
INSERT INTO dbo.Test VALUES (5, """"Text in double quotes""")
```

```
INSERT INTO dbo.Test VALUES (6, "Text with 2 """" double quotes")
```

```
GO
```

```
select * from Test
```

Các tham số - SET NOEXEC { ON | OFF }

SET NOEXEC On

--không thực hiện chương trình mà chỉ biên dịch

print 'valid on'

go

SET NOEXEC Off -- default

-- biên dịch và thực hiện chương trình

print 'valid off'

go

Các tham số - DATEFORMAT

Định dạng ngày tháng

```
SET DATEFORMAT mdy
```

```
GO
```

```
DECLARE @datevar datetime
```

```
SET @datevar = '12/31/98'
```

```
SELECT @datevar
```

```
GO
```

```
SET DATEFORMAT ydm
```

```
GO
```

```
DECLARE @datevar datetime
```

```
SET @datevar = '98/31/12'
```

```
SELECT @datevar
```

```
GO
```

```
SET DATEFORMAT ymd
```

```
GO
```

```
DECLARE @datevar datetime
```

```
SET @datevar = '98/12/31'
```

```
SELECT @datevarGO
```

Các tham số - SET NOCOUNT { ON | OFF }

```
USE AdventureWorks;GO
```

```
SET NOCOUNT OFF;GO
```

```
-- Display the count message.
```

```
SELECT TOP(5) LastName
```

```
FROM Person.Contact
```

```
WHERE LastName LIKE 'A%';GO
```

```
-- SET NOCOUNT to ON to no display the count message.
```

```
SET NOCOUNT ON;
```

```
GO
```

```
SELECT TOP(5) LastName
```

```
FROM Person.Contact
```

```
WHERE LastName LIKE 'A%'; GO
```

```
-- Reset SET NOCOUNT to OFF
```

```
SET NOCOUNT OFF;GO
```


Hàm USER_NAME()

USER_NAME – trả về tên user hiện thời; với các user thuộc nhóm **sysadmin fixed server role** thì trả về dbo.

```
CREATE DATABASE TestCOC
```

```
go
```

```
USE TestCOC
```

```
go
```

```
SELECT USER_NAME();
```

```
GO
```

```
EXECUTE AS USER = 'hung';
```

```
GO
```

```
SELECT USER_NAME();
```

```
GO
```

```
REVERT;GO
```

```
SELECT USER_NAME();GO
```