

BÀI GIẢNG QUẢN TRỊ CƠ SỞ DỮ LIỆU

hoint2002@gmail.com or hoint@tmu.edu.vn
Bộ môn CNTT – Khoa HTTT Kinh tế&TMĐT

NỘI DUNG HỌC PHẦN

- Phần 1: Tổng quan về quản trị CSDL
- Phần 2: Hệ quản trị CSDL SQL Server
- Phần 3: Các hoạt động quản trị CSDL trong hệ thống thông tin

BÀI 01

TỔNG QUAN VỀ QUẢN TRỊ CSDL

NỘI DUNG BÀI 1

- TỔNG QUAN VỀ QUẢN TRỊ CSDL
 - Hoạt động quản trị cơ sở dữ liệu
 - Người quản trị cơ sở dữ liệu
 - Phần mềm quản trị cơ sở dữ liệu
 - Kiến trúc hệ quản trị cơ sở dữ liệu

1.1. Hoạt động quản trị CSDL

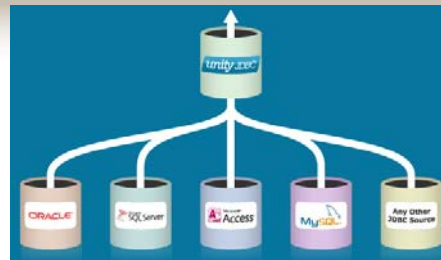
- Khái niệm CSDL:
 - Là một tập hợp dữ liệu có cấu trúc được lưu trên các thiết bị lưu trữ như băng từ, đĩa từ nhằm thỏa mãn nhu cầu khai thác đồng thời của nhiều người sử dụng, nhiều chương trình ứng dụng tại cùng thời điểm
 - CSDL là một bộ sưu tập các dữ liệu tác nghiệp được lưu trữ lại và được các hệ ứng dụng của một đơn vị, tổ chức nào đó sử dụng

1.1. Hoạt động quản trị CSDL Các kiểu dữ liệu phổ biến

Data	Represented by
Alphanumeric data	Numbers, letters, and other characters
Image data	Graphic images or pictures
Audio data	Sound, noise, tones
Video data	Moving images or pictures

1.1. Hoạt động quản trị CSDL

- Ưu điểm của CSDL
 - Giảm sự trùng lặp thông tin -> đảm bảo tính nhất quán và toàn vẹn dữ liệu
 - Dữ liệu có thể truy xuất theo nhiều cách khác nhau mà không mất mát thông tin
 - Có khả năng chia sẻ thông tin



1.1. Hoạt động quản trị CSDL

- Một số vấn đề đặt ra khi sử dụng CSDL
 - Làm thế nào để đảm bảo được tính chủ quyền của dữ liệu -> Cần có biện pháp an toàn dữ liệu, có các ràng buộc nhằm toàn vẹn ngữ nghĩa, đảm bảo toàn vẹn dữ liệu
 - Vấn đề bảo mật và quyền của người sử dụng trong môi trường mạng
 - Giải quyết tranh chấp dữ liệu khi nhiều người dùng cùng sử dụng dữ liệu -> quản trị giao dịch tốt
 - Vấn đề an toàn, phòng chống khả năng phục hồi ?

1.1. Hoạt động quản trị CSDL



1.1. Hoạt động quản trị CSDL

- Vai trò của quản trị CSDL trong HTTT
 - Có vai trò quan trọng do
 - Dữ liệu là một loại tài sản đặc biệt của các tổ chức, đơn vị kinh doanh
 - Nếu tổ chức, đơn vị kinh doanh có hoạt động quản trị CSDL tốt thì năng suất HTTT sẽ hiệu quả và giảm thiểu rủi ro cho hệ thống

1.1. Hoạt động quản trị CSDL

- Mục đích của QTCSDL
 - Đảm bảo CSDL luôn luôn toàn vẹn, bí mật và sẵn sàng cho người sử dụng trong hệ thống
 - Các công việc cần thực hiện
 - Theo dõi định kỳ, xử lý sự cố
 - Sao lưu
 - Điều chỉnh CSDL
 - Bảo mật
 - Phục hồi
 - ...

1.2 Người quản trị CSDL

- Khái niệm
 - DBA (DataBase Administrator): Là người có trách nhiệm điều khiển tập trung đối với dữ liệu cũng như các chương trình, người truy cập đến dữ liệu
 - DBA là người có trách nhiệm cài đặt, vận hành, duy trì, kiểm soát, sao lưu và xử lý sự cố cho CSDL của hệ thống

Một số nhiệm vụ của DBA Oracle

- Task 1: Evaluate the Database Server Hardware
- Task 2: Install the Oracle Database Software
- Task 3: Plan the Database
- Task 4: Create and Open the Database
- Task 5: Back Up the Database
- Task 6: Enroll System Users
- Task 7: Implement the Database Design
- Task 8: Back Up the Fully Functional Database
- Task 9: Tune Database Performance
- Task 10: Download and Install Patches
- Task 11: Roll Out to Additional Hosts

1.2 Người quản trị CSDL Một số nhiệm vụ của DBA

- Xây dựng CSDL:
 - Thiết kế các lược đồ vật lý và lược đồ khái niệm
 - Tinh chỉnh CSDL
 - Thiết kế các ràng buộc và toàn vẹn dữ liệu
- Đảm bảo an toàn và bảo mật
 - Xác định các nguy cơ
 - Lựa chọn, cài đặt các biện pháp phòng chống và khắc phục sự cố
 - Đảm bảo xác thực và toàn vẹn

1.2 Người quản trị CSDL Một số nhiệm vụ của DBA

- Quản lý vận hành
 - Đảm bảo tính sẵn sàng
 - Xử lý các truy vấn và giao dịch
 - Cải thiện hệ thống xử lý
 - Tối ưu tổ chức lưu trữ
 - Lựa chọn hệ QTCSDL
 - Cài đặt, nâng cấp khi cần thiết

1.2 Người quản trị CSDL Một số yêu cầu đối với DBA

- Có hiểu biết về HT CSDL và HTTT của tổ chức
- Có kiến thức về CSDL, CNTT, HT mạng máy tính
- Có kỹ năng giao tiếp, hiểu và phân tích được yêu cầu của khách hàng và người quản lý
- Nhanh nhạy nắm bắt các công nghệ mới trong quản trị CSDL và các biện pháp an ninh cho HT CSDL

1.3 Phần mềm quản trị CSDL

- Lý do ra đời
 - Kích thước và độ phức tạp của CSDL ngày càng lớn
 - Dung lượng lưu trữ, thời gian truy xuất càng ngày càng đòi hỏi nhiều hơn
 - Các nguy cơ mất an toàn đối với dữ liệu ngày càng tăng nhanh
 - Trình độ và khả năng sử dụng CSDL của người dùng rất khác nhau
- => Cần có phần mềm hỗ trợ người dùng trong tổ chức, truy cập, truy xuất, ...

1.3 Phần mềm quản trị CSDL

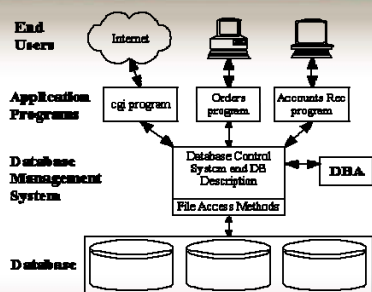
- Dung lượng khổng lồ
- Độ phức tạp lớn
- Độ an toàn thấp
- Yêu cầu của người dùng



1.3 Phần mềm quản trị CSDL Hệ quản trị CSDL

- Khái niệm
 - Là một phần mềm cho phép tạo lập, lưu trữ và truy xuất thông tin từ một CSDL
 - Thông thường một hệ QT CSDL có thể hỗ trợ cho nhiều mô hình dữ liệu khác nhau

Hệ QT CSDL trong HTTT



1.3 Phần mềm quản trị CSDL Đặc điểm của DBMS

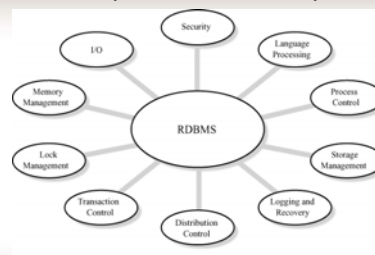
- Quản trị giao dịch
- Điều khiển truy cập
- Toàn vẹn dữ liệu
- Phục hồi dữ liệu

1.3 Phần mềm quản trị CSDL Đặc điểm của DBMS

- Hỗ trợ ít nhất một mô hình dữ liệu:
 - Hỗ trợ mô hình dữ liệu phân cấp: IMS (IBM)
 - Hỗ trợ mô hình dữ liệu quan hệ: System_R (IBM), DB2, Bbase, Oracle, SQL Server, My SQL, ...
 - Hỗ trợ mô hình dữ liệu hướng đối tượng: Orien, Itasca, ODMG, O2, ...
- Đảm bảo tính độc lập dữ liệu về mặt vật lý và logic
- Hỗ trợ các ngôn ngữ cấp cao nhất định
 - My SQL (asp, php), SQL Server (.net, ...)...

1.3 Phần mềm quản trị CSDL Hệ QT CSDL quan hệ - RDBMS

- Là hệ QT CSDL hỗ trợ cho mô hình dữ liệu quan hệ
- Vai trò



1.4 Kiến trúc hệ quản trị CSDL Thao tác trong hệ QT CSDL

- Tạo mới, thay đổi, chỉnh sửa lược đồ
- Cập nhật dữ liệu gồm:
 - Thêm mới, chỉnh sửa, lưu, xóa
- Truy vấn dữ liệu
 - Thông qua một giao diện truy vấn chung
 - Thông qua các chương trình ứng dụng

Ví dụ

Tổng kết

- Khái niệm CSDL? Quản trị CSDL? Hệ QT CSDL? Kiến trúc hệ CSDL?
- Mục đích của hệ QTCSDL? Vai trò của QT CSDL trong hệ thống?
- Những vấn đề gặp phải trong quá trình sử dụng CSDL?

BÀI 02

HỆ QUẢN TRỊ CSDL SQL SERVER

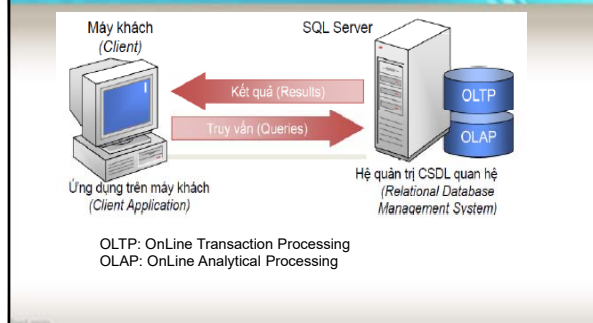
NỘI DUNG BÀI 2

- **HỆ QUẢN TRỊ CSDL SQL SERVER**
 - Tổng quan về SQL Server
 - Mô hình hoạt động của SQL Server trong HT mạng
 - Một số công cụ của SQL Server 2008

2.1 TỔNG QUAN VỀ SQL SERVER

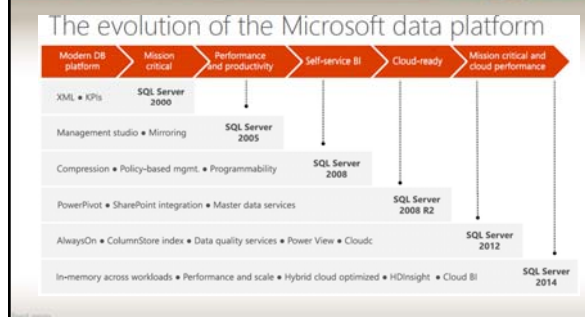
- **Vài nét về SQL Server**
 - Là hệ QT CSDL hỗ trợ mô hình dữ liệu quan hệ do Microsoft phát triển
 - Hoạt động hỗ trợ mô hình Khách/Chủ (Client/Server)

2.1 TỔNG QUAN VỀ SQL SERVER



2.1 TỔNG QUAN VỀ SQL SERVER

Lịch sử



2.1 TỔNG QUAN VỀ SQL SERVER

Lịch sử

FEATURES	SQL Server 2014	SQL Server 2012	SQL Server 2008 R2	SQL Server 2008
Performance				
In-Memory OLTP*	✓	✗	✗	✗
In-Memory ColumnStore*	✓	✗	✗	✗
Buffer Pool Extension to SSDs	✓	✗	✗	✗
Enhanced virtualization support and Live Migration	✓	✓	✓	✗
Availability				
Always On*	✓	✓	✗	✗
Database Recovery Advisor	✓	✓	✗	✗
Windows Server Core Support	✓	✓	✗	✗
Security				
Transparent Data Encryption*	✓	✓	✓	✓
Backup Encryption Support	✓	✗	✗	✗
Enhanced Separation of Duties	✓	✗	✗	✗
Cloud-Readiness				
Simplified Backup to Microsoft Azure	✓	✗	✗	✗
Larger SQL Server VMs and memory sizes available in Microsoft Azure	✓	✗	✗	✗
Gallery of VM images in Microsoft Azure	✓	✓	✓	✗
Management & Programmability				
Policy-Based Management	✓	✓	✓	✓
Distributed Replay	✓	✓	✗	✗

*SQL Server Enterprise Edition only

2.1 TỔNG QUAN VỀ SQL SERVER

Giới thiệu

- Vai trò của RDBMS
 - Duy trì các quan hệ chứa dữ liệu trong CSDL
 - Bảo đảm dữ liệu được lưu trữ đúng và hợp lệ
 - Có khả năng phục hồi dữ liệu

2.1 TỔNG QUAN VỀ SQL SERVER

Giới thiệu

- SQL Server
 - Quản lý truy cập và phân quyền người dùng
 - Quản lý tập trung
 - Có hỗ trợ OLAP và OLTP
 - Ứng dụng được cho TMDT và Kho dữ liệu
 - Sử dụng T_SQL
 - Có DDL (Data Definition Language)
 - Có DML (Data Manipulation Language)
 - Có DCL (Database Control Language)

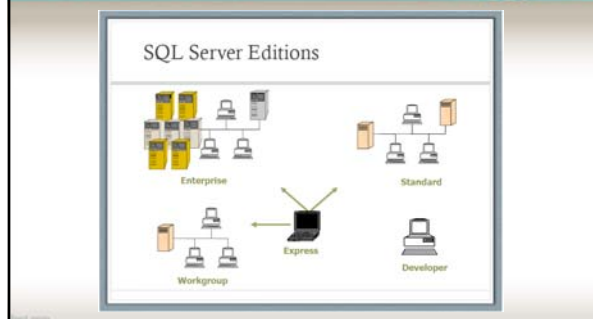
2.1 TỔNG QUAN VỀ SQL SERVER

Giới thiệu SQL Server 2008

- Truy cập tới CSDL qua ht mạng
- Hỗ trợ mô hình Client/Server
- Kho dữ liệu (Data Warehouse)
- Tương thích với chuẩn ANSI/ISO SQL-92
- Hỗ trợ tìm kiếm full-text
- Hỗ trợ tìm kiếm trực tuyến (Books Online)
- Hỗ trợ nhiều kiểu dữ liệu mới: XML, pictures, video, ...
- Hỗ trợ File Stream
- Hỗ trợ .Net 3.5

2.1 TỔNG QUAN VỀ SQL SERVER

Giới thiệu SQL Server 2008

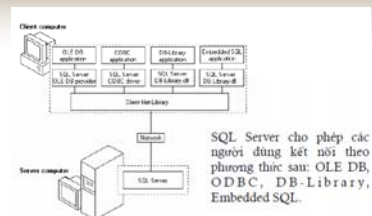


2.2. Mô hình hoạt động

Mô hình Client/Server

Client:
Đóng vai trò
business logic và
biểu diễn dữ
liệu

Server:
Thực hiện các
tiến trình để
quản lý CSDL,
bảo mật, thực
hiện truy vấn,
....

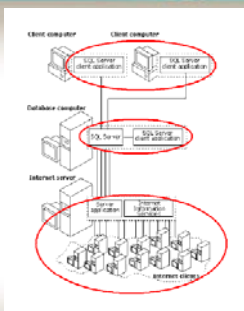


SQL Server cho phép các
người dùng kết nối theo
phương thức sau: OLE DB,
ODBC, DB-Library,
Embedded SQL.

2.2. Mô hình hoạt động

Mô hình SQL trên mạng

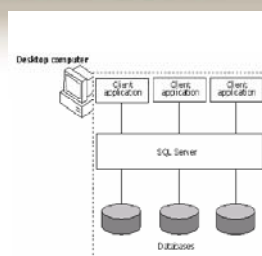
- Có thể thực hiện trao đổi dữ liệu trên nhiều mô hình mạng, nhiều giao thức, nhiều phương thức truyền tin khác nhau



2.2. Mô hình hoạt động

Mô hình Desktop

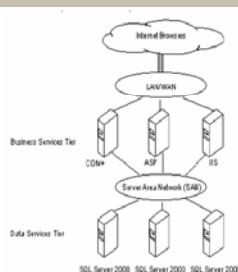
- Trên một máy Desktop có thể có nhiều ứng dụng, mỗi ứng dụng có thể thực hiện thao tác với nhiều CSDL.



2.2. Mô hình hoạt động

Mô hình kết nối ứng dụng

- Các máy chủ SQL Server sẽ được quản lý thông qua các hệ thống máy chủ mạng, hệ điều hành mạng.
- Các ứng dụng (COM+, ASP, IIS) sẽ thông qua máy chủ mạng kết nối đến SQL Server.
- Mô hình áp dụng cho các mạng nội bộ, diện rộng.
- Ứng dụng được khai thác trên trình duyệt Internet Browser.



2.3 Một số công cụ của SQL Server 2008

Công cụ	Nội dung
SQL Server Configuration Management	Khởi động và dừng Server dữ liệu
SQL Server Management Studio	Thao tác với CSDL
Books Online	Bộ tài liệu

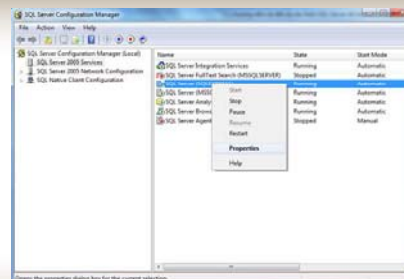
2.3 Một số công cụ của SQL Server 2008 SQL Server Configuration Management

- Công cụ trực quan hỗ trợ quản trị dịch vụ trên Server
- Hỗ trợ thiết lập nhiều máy tính liên kết với nhau để trao đổi dữ liệu
- Hỗ trợ thiết lập các Instance trên máy đơn
- Hỗ trợ một số tác vụ

sapt.com

2.3 Một số công cụ của SQL Server 2008 SQL Server Configuration Management

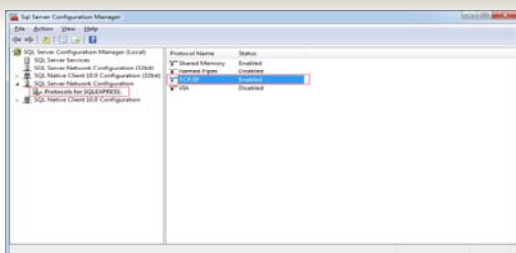
- Các tác vụ thường gặp



sapt.com

2.3 Một số công cụ của SQL Server 2008 SQL Server Configuration Management

- Cấu hình kết nối mạng



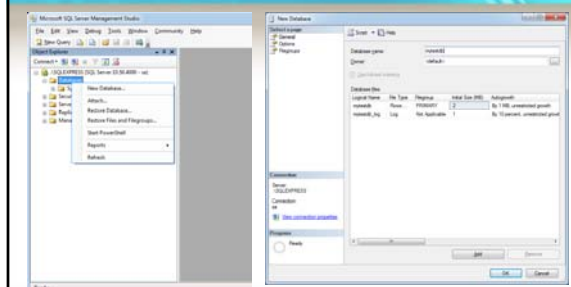
sapt.com

2.3 Một số công cụ của SQL Server 2008 SQL Server Management Studio

- Công cụ chính để quản lý, thiết kế và xây dựng CSDL
- Kết hợp Enterprise Manager và Query Analyzer của SQL Server 2000
- Giao diện trực quan, dễ dùng, dễ điều khiển

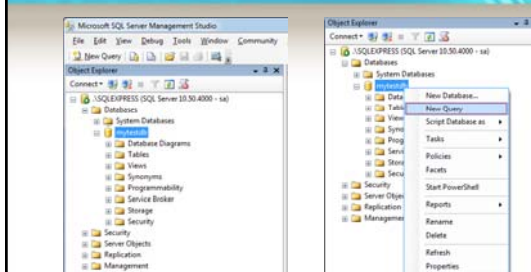
sapt.com

2.3 Một số công cụ của SQL Server 2008 SQL Server Management Studio Tạo CSDL mới



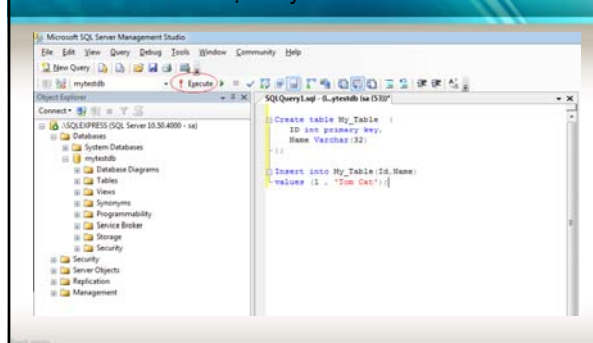
sapt.com

2.3 Một số công cụ của SQL Server 2008 SQL Server Management Studio Tạo truy vấn mới

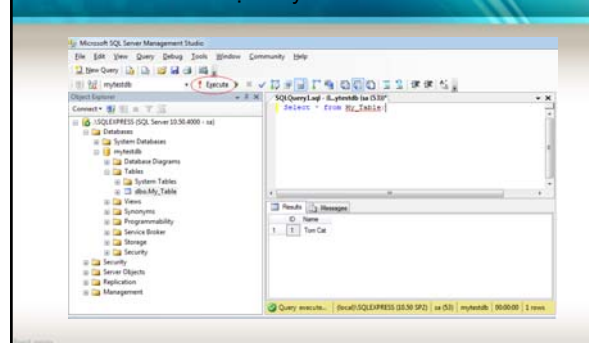


sapt.com

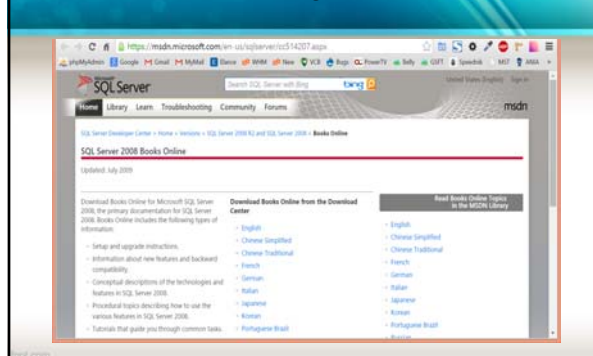
2.3 Một số công cụ của SQL Server 2008 SQL Server Management Studio Tạo truy vấn mới



2.3 Một số công cụ của SQL Server 2008 SQL Server Management Studio Tạo truy vấn mới



2.3 Một số công cụ của SQL Server 2008 Books Online



Tổng kết

- Khái niệm SQL Server ? Lịch sử phát triển? Những đặc trưng và cải tiến của phiên bản mới nhất
- Các công cụ trong SQL Server 2008? So sánh SQL server và Oracle, DB2, My SQL, ...
- Cài đặt SQL Server 2008, thiết lập CSDL theo bài tập giáo viên cho

BÀI 03

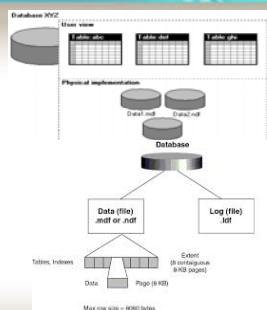
THAO TÁC VỚI CSDL TRONG SQL SERVER

NỘI DUNG BÀI 3

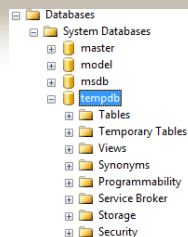
- THAO TÁC VỚI CSDL TRONG SQL SERVER
 - Cấu trúc CSDL SQL Server
 - Tạo CSDL và thao tác với bảng
 - Sao chép, gán/tách, nhập/xuất CSDL

3.1 Cấu trúc CSDL trong SQL Server Cơ chế lưu trữ

- Mỗi một Database lưu thành 2 phần
 - Tập tin gồm File.mdf và các tập tin phụ File.ndf
 - Tập tin nhật ký File.ldf



3.1 Cấu trúc CSDL trong SQL Server System Database



- Master: Lưu toàn bộ thông tin HT của SQL Server: TT đăng nhập, cấu hình HT, ...
- Model: Mô hình dùng chung cho tất cả các CSDL xây dựng trong một Instance
- Msdb: Do SQL Server Agent quản lý để điều khiển hoạt động của CSDL
- Tempdb: CSDL dùng chung cho mọi người dùng sử dụng trong một Instance gồm bảng, query, procedured, ...

3.1 Cấu trúc CSDL trong SQL Server Objects

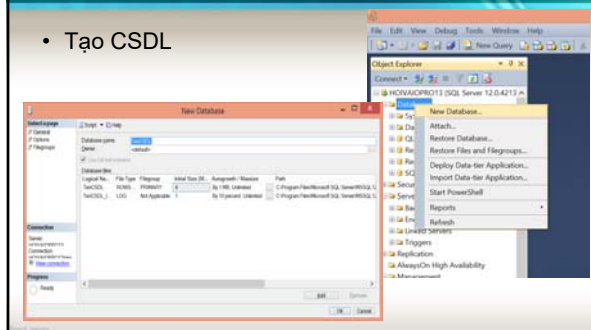
- Các bảng dữ liệu (Tables)
- Các chỉ mục (Index)
- Các khung nhìn (Views)
- Các thủ tục lưu trữ (Store Procedures)
- Các bẫy lỗi (Triggers)

3.1 Cấu trúc CSDL trong SQL Server

- SQL Server quản lý trực tiếp các CSDL. Mỗi CSDL sẽ quản lý các cấu trúc vật lý của nó. Do đó:
 - Client phải kết nối đến Server để khai thác CSDL
 - Các Client chỉ khai thác được các CSDL có tên trong danh sách mà Server quản lý
 - Không mở CSDL trực tiếp được
 - Client chỉ thực hiện được các quyền cho phép

3.2 Tạo CSDL và thao tác với bảng Tạo mới CSDL

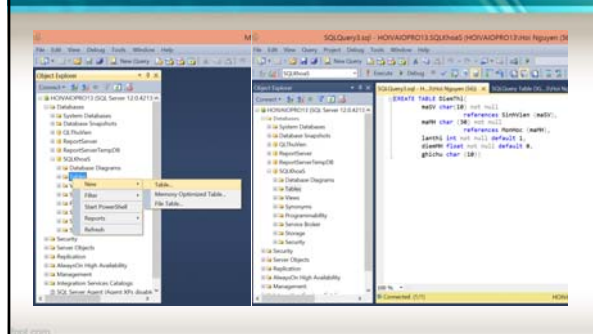
- Tạo CSDL



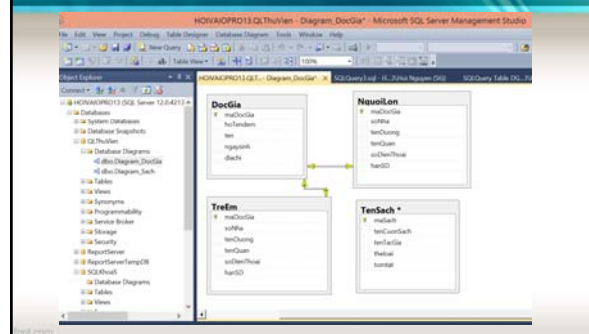
3.2 Tạo CSDL và thao tác với bảng Các kiểu dữ liệu

DATA TYPE	DESCRIPTION
bigint	Integer data from -2 ⁶³ through 2 ⁶³ -1
int	Integer data from -2 ³¹ through 2 ³¹ -1
smallint	Integer data from -2 ¹⁵ through 2 ¹⁵ -1
tinyint	Integer data from 0 through 255
bit	Integer data with either a 1 or 0 value
decimal	Fixed precision and scale numeric data from -10 ³⁸ +1 through 10 ³⁸ -1
numeric	Fixed precision and scale numeric data from -10 ³⁸ +1 through 10 ³⁸ -1
money	Monetary data values from -2 ⁶³ through 2 ⁶³ -1
smallmoney	Monetary data values from -214,748,364.8 through 214,748,364.7
float	Floating precision number data from -1.79E+308 through 1.79E+308
real	Floating precision number data from -3.40E+38 through 3.40E+38
datetime	Date and time data from January 1, 1900, through December 31, 9999, with an accuracy of 3.33 milliseconds
smalldatetime	Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute
char	Fixed-length character data with a maximum length of 8,000 characters
varchar	Variable-length data with a maximum of 8,000 characters
text	Variable-length data with a maximum length of 2 ³¹ -1 characters
nchar	Fixed-length Unicode data with a maximum length of 4,000 characters
nvarchar	Variable-length Unicode data with a maximum length of 4,000 characters

3.2 Tạo CSDL và thao tác với bảng Tạo bảng/Tạo truy vấn



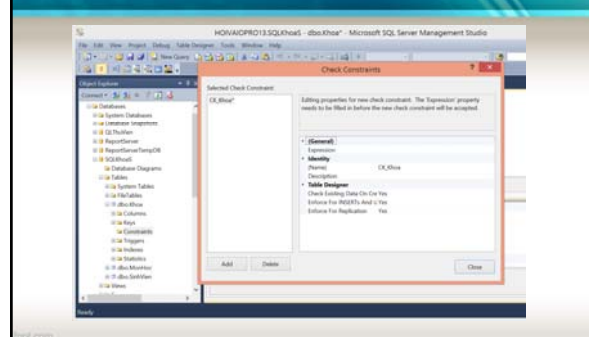
3.2 Tạo CSDL và thao tác với bảng Ràng buộc và Quan hệ



3.2 Tạo CSDL và thao tác với bảng Ràng buộc và Quan hệ

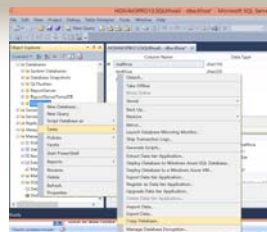
- Ràng buộc
 - Ràng buộc cho cột (trường): Not null, references, ...
 - Ràng buộc cho bảng: Not null, Check, Unique, Primary Key, Foreign Key,

3.2 Tạo CSDL và thao tác với bảng Thiết kế ràng buộc

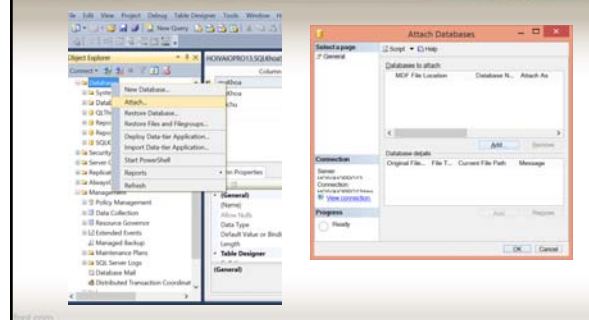


3.3 Thao tác với CSDL Sao chép

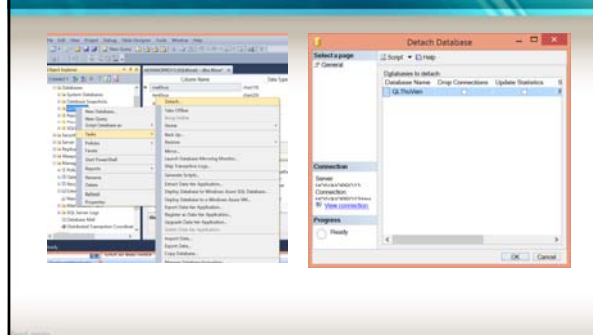
- Sao chép CSDL cho phép copy và remove CSDL từ một instance này đến một instance khác



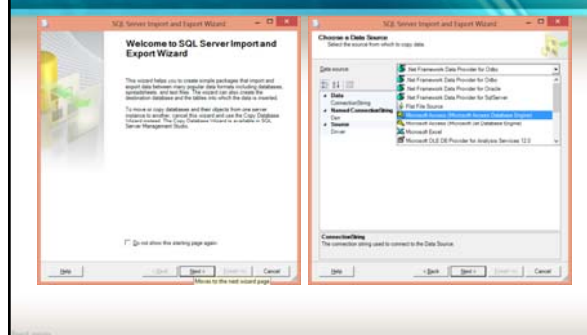
3.3 Thao tác với CSDL Gán (Attach)



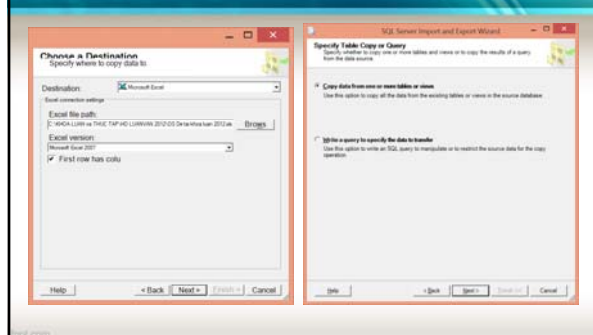
3.3 Thao tác với CSDL Tách (Detach)



3.3 Thao tác với CSDL Nhập (Import)



3.3 Thao tác với CSDL Xuất (Export)



Tổng kết

- Cài đặt SQL Server 2008
- Các thao tác với CSDL, với bảng, các ràng buộc
- Các thao tác quản trị CSDL, sao chép, gán, tách, nhập, xuất, ...
- Học lại các câu lệnh truy vấn tạo bảng, tạo các quan hệ giữa các bảng và các nội dung truy vấn khác

BÀI 04

NGÔN NGỮ TRUY VẤN T_SQL TRONG SQL SERVER

NỘI DUNG BÀI 4

- NGÔN NGỮ TRUY VẤN T_SQL
 - Các câu lệnh cơ bản
 - Các kiểu biến
 - Các cấu trúc điều khiển

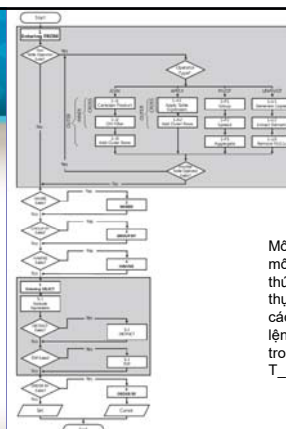
NGÔN NGỮ TRUY VẤN T_SQL

Thứ tự thực hiện

- Thứ tự thực hiện trong câu truy vấn T-SQL như sau:
 - Mệnh đề FROM sẽ được thực hiện đầu tiên
 - Sau đó lần lượt là các phép toán giữa các bảng (table) như JOIN, APPLY, PIVOT, UNPIVOT...
 - Tiếp theo là mệnh đề WHERE
 - Tới mệnh đề ORDER cuối cùng.
 - Tại mỗi bước thực hiện xong thì kết quả trả về là một bảng ảo (virtual table).

```
(5) SELECT (5-2) DISTINCT (5-3) TOP(<top_specification>) (5-1) <select_list>
(1) FROM (1-2) <left_table> <join_type> JOIN <right_table> ON <on_predicate>
    | (1-4) <left_table> <apply_type> APPLY <right_table_expression> AS <alias>
    | (1-P) <left_table> PIVOT(<pivot_specification>) AS <alias>
    | (1-U) <left_table> UNPIVOT(<unpivot_specification>) AS <alias>
(2) WHERE <where_predicate>
(3) GROUP BY <group_by_specification>
(4) HAVING <having_predicate>
(6) ORDER BY <order_by_list>;
```

ngl.com



Mô hình
mô tả
thứ tự
thực hiện
các câu
lệnh
trong
T_SQL

ngl.com

4.1 Các câu lệnh cơ bản Ngôn ngữ định nghĩa dữ liệu

TẠO CÁC ĐỐI TƯỢNG

Câu lệnh	Mô tả
CREATE DATABASE	Tạo mới một CSDL
CREATE TABLE	Tạo mới một bảng trong CSDL hiện tại
CREATE INDEX	Tạo một chỉ mục mới cho bảng được chỉ định trước
CREATE FUNCTION	Tạo hàm người dùng định nghĩa
CREATE PROCEDURE	Tạo thủ tục lưu trữ
CREATE TRIGGER	Tạo mới một kích hoạt
CREATE VIEW	Tạo mới một khung nhìn

ngl.com

Ví dụ tạo bảng

```
Create table [Tên Bảng]
(
    [Thuộc tính 1] [Kiểu dữ liệu 1],
    [Thuộc tính 2] [Kiểu dữ liệu 2],
    ...
    [Thuộc tính n] [Kiểu dữ liệu n],
    CONSTRAINT [Tên Khóa Chính]
    PRIMARY KEY ([Danh sách các thuộc tính
    làm khóa chính, cách nhau bằng dấu
    phẩy])
    CONSTRAINT [Tên Khóa Ngoại]
    FOREIGN KEY ([Danh sách các thuộc tính
    khóa ngoại]) REFERENCES [Tên bảng tham
    chiếu]([Danh sách các thuộc tính khóa
    chính của bảng tham chiếu đến])
)
```

ngl.com

4.1 Các câu lệnh cơ bản Ngôn ngữ định nghĩa dữ liệu

SỬA ĐỔI CẤU TRÚC CÁC ĐỐI TƯỢNG

Câu lệnh	Mô tả
ALTER TABLE	Sửa thiết kế bảng
ALTER TRIGGER	Sửa kích hoạt
ALTER VIEW	Sửa khung nhìn
ALTER PROCEDURE	Sửa thủ tục lưu trữ
ALTER FUNCTION	Sửa hàm người dùng định nghĩa

ngl.com

Ví dụ sửa đổi cấu trúc bảng

<u>Thêm thuộc tính i</u> Alter table [Tên Bảng] add [Tên thuộc tính] [Kiểu dữ liệu]	<u>Thêm thuộc tính ĐịaChị</u> Alter table NhanVien add ĐịaChị varchar(20)
<u>Xóa thuộc tính i</u> Alter table [Tên Bảng] drop column [Tên thuộc tính]	<u>Xóa thuộc tính ĐịaChị</u> Alter table NhanVien drop ĐịaChị
<u>Sửa thuộc tính i</u> Alter table [Tên Bảng] alter column [Tên thuộc tính] [Kiểu dữ liệu mới]	<u>Sửa thuộc tính ĐịaChị</u> Alter table NhanVien Alter column ĐịaChị varchar(50)

ngl.com

4.1 Các câu lệnh cơ bản Ngôn ngữ định nghĩa dữ liệu

XÓA CÁC ĐỐI TƯỢNG

Câu lệnh	Mô tả
DROP DATABASE	Xóa một CSDL
DROP TABLE	Xóa bảng
DROP INDEX	Xóa một chỉ mục
DROP FUNCTION	Xóa hàm người dùng định nghĩa
DROP PROCEDURE	Xóa một thủ tục lưu trữ
DROP TRIGGER	Xóa một kích hoạt
DROP VIEW	Xóa một khung nhìn

Ví dụ xóa

```
Alter table [Tên Bảng] drop constraint
[Tên ràng buộc]
```

Xóa khóa chính

```
Alter table NHANVIEN drop constraint
FK_NHANVIEN
```

Xóa khóa ngoại

```
Alter table PHONG drop constraint
FK_PHONG_NHANVIEN
```

4.1 Các câu lệnh cơ bản Ngôn ngữ thao tác dữ liệu

CÁC CÂU LỆNH CƠ BẢN THAO TÁC VỚI DỮ LIỆU

Câu lệnh	Mô tả
SELECT	Truy xuất dữ liệu từ một hoặc nhiều bảng
INSERT	Thêm một hàng dữ liệu vào bảng
UPDATE	Cập nhật dữ liệu
DELETE	Xóa dữ liệu

Ví dụ

```
Xem nội dung của một bảng
SELECT * FROM [Tên bảng]

Xóa nội dung của một bảng
DELETE FROM [Tên bảng]
WHERE [Biểu thức điều kiện]
```

Xem dữ liệu của bảng NHANVIEN

```
SELECT * FROM NHANVIEN
```

Xóa nhân viên NV001 của bảng NHANVIEN

```
DELETE FROM NHANVIEN
```

```
WHERE manv = 'NV001'
```

Xóa tất cả dữ liệu bảng NHANVIEN

```
DELETE FROM NHANVIEN
```

Cho biết các nhân viên sinh trong khoảng năm 1955 đến 1960

```
SELECT *
FROM NHANVIEN
WHERE NG SINH between '1/1/1955' and '12/31/1960'
```

4.1 Các câu lệnh cơ bản Ngôn ngữ điều khiển dữ liệu

CÁC CÂU ĐIỀU KHIỂN DỮ LIỆU

Câu lệnh	Mô tả
GRANT	Gán quyền người dùng lên các đối tượng
REVOKE	Xóa quyền người dùng lên các đối tượng
DENY	Ngăn quyền người dùng lên các đối tượng

4.2 Các kiểu biến trong T_SQL Biến

- Biến
 - Biến là một đối tượng để lưu trữ một giá trị dữ liệu. Có thể dùng câu lệnh T_SQL để truyền dữ liệu thông qua các tên biến
 - Biến thường có 2 loại
 - Biến cục bộ: Khai báo và dùng tạm thời khi thực hiện câu lệnh T_SQL. Bắt đầu bằng @
 - Biến toàn cục: Khai báo trước và do hệ thống quản lý. Bắt đầu bằng @@

4.2 Các kiểu biến trong T_SQL

Biến vô hướng

- Biến vô hướng
 - Dùng lưu trữ các giá trị dữ liệu chuẩn mà SQL hỗ trợ. Các giá trị đơn có thể khai báo trước khi sử dụng
 - Khai báo và sử dụng
 - DECLARE @[Tên biến] [Kiểu dữ liệu]
- ```
DECLARE @OrderId int;
SET @OrderId = 10000;
SELECT * FROM Orders
WHERE OrderID > @OrderId;
```

## 4.2 Các kiểu biến trong T\_SQL

### Biến vô hướng

#### Ví dụ: Gán giá trị biến bằng SET

```
--Thức gọi: sp_? (Ví dụ: sp_? là SP_? trong SQL Server)
USE VietnameseBusinessZone
GO
-- Khai báo biến
DECLARE @VendorIDVar int -- Giá trị ID của nhà cung cấp
-- Gán giá trị cho biến @VendorIDVar
SET @VendorIDVar = 1
SELECT * FROM CompanyProducts
WHERE CompanyID = @VendorIDVar
GO
```

| ProductID | ProductName     | CompanyID | Supplier        | Keywords                   | CategoryID | Unit | DisplayPrice |
|-----------|-----------------|-----------|-----------------|----------------------------|------------|------|--------------|
| 1         | SQL Server 2000 | 1         | SQL Server 2000 | SQL Server, SQL, 2000      | 1          | Cuốn | 900          |
| 2         | SQL Server 2005 | 1         | SQL Server 2005 | SQL Server, SQL, 2005      | 1          | Cuốn | 900          |
| 3         | SQL Server 2008 | 1         | SQL Server 2008 | SQL, SQL Server, SQL, 2008 | 1          | Cuốn | 900          |

## 4.2 Các kiểu biến trong T\_SQL

### Biến bảng

- Biến bảng
    - Một bảng được xem như một biến, sẽ truyền cả nội dung bảng khi thực hiện câu lệnh T\_SQL
    - Khai báo và sử dụng
      - DECLARE @<tableVariableName> TABLE (<column1 definition> [, ...])
- ```
CREATE PROCEDURE spTableVariableTest AS
DECLARE @myTable TABLE (col1 INT, col2 varchar(20))
INSERT INTO @myTable (col1, col2) VALUES (10, 'Monday')
INSERT INTO @myTable (col1, col2) VALUES (22, 'Tuesday')
SELECT *
FROM @myTable
GO
```

4.3 Các cấu trúc điều khiển

- Các cấu trúc điều khiển được sử dụng khi việc xử lý dữ liệu có nhiều điều kiện ràng buộc cần thỏa mãn
- Thường phân thành 2 loại
 - Rẽ nhánh: IF, Case, ...
 - Lặp: While, Repeat, ...

4.3 Các cấu trúc điều khiển

IF ... ELSE

```
IF biểu_thức_điều_kiện BEGIN
    các_lệnh_xử_ly_khi_thỏa_điều_kiện
END
ELSE BEGIN
    các_lệnh_xử_ly_khi_không_thỏa_điều_kiện
END

DECLARE @soLuong INT
SELECT @soLuong = COUNT(*) FROM SINHVIEN

IF @soLuong > 0 BEGIN
    PRINT @soLuong
END
ELSE BEGIN
    PRINT N'Không có sinh viên'
END
```

4.3 Các cấu trúc điều khiển

IF ... ELSE lồng nhau

```
FUNCTION TAX_INC_M (chuoi VARCHAR(63)) RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @TEMP DECIMAL(10,2)
    IF (@INCOME > 1500)
        SET @TEMP = @INCOME * 0.15
    ELSE IF (@INCOME > 1000)
        SET @TEMP = @INCOME * 0.1
    ELSE IF (@INCOME > 500)
        SET @TEMP = @INCOME * 0.05
    ELSE
        SET @TEMP = 0
    RETURN @TEMP
END
```

ENO	ENAME	INCOME	TAX
1	E0001	THA	465 0.00
2	E0002	QU	1620 243.00
3	E0003	THANH	870 43.50

4.3 Các cấu trúc điều khiển

Cấu trúc While

Cú pháp:

```
WHILE biểu_thức_điều_kiện BEGIN
    các_lệnh_xử_ly_khi_thỏa_điều_kiện
END
```

Ví dụ:

```
DECLARE @i INT
SET @i = 1

WHILE @i < 100 BEGIN
    PRINT @i
    SET @i = @i + 1
END
```

4.3 Các cấu trúc điều khiển

Cấu trúc Case 1

A. Cấu trúc 1

```
CASE input_expression
WHEN when_expression THEN result_expression
[ ...n ]
[ ELSE else_result_expression
]
```

Vi dụ:

- Cho biết họ tên các nhân viên và năm về hưu

```
SELECT HỌ_TÊN, TENNV,
(CASE PHAI
WHEN 'Nam' THEN YEAR(NGSIINH) + 60
WHEN 'Nu' THEN YEAR(NGSIINH) + 55
END) AS NAMVIEU
FROM NHANVIEN
```

4.3 Các cấu trúc điều khiển

Cấu trúc Case 2

B. Cấu trúc 2

```
CASE
WHEN Boolean_expression THEN result_expression
[ ...n ]
[ ELSE else_result_expression
]
```

Vi dụ:

- Cho biết sinh viên và xếp loại học lực của sinh viên

```
select MASV, CASE
WHEN DIEMTB >= 8 THEN N'Giỏi'
WHEN DIEMTB >= 7 THEN N'Khá'
WHEN DIEMTB >= 6 THEN N'Trung bình khá'
WHEN DIEMTB >= 5 THEN N'Trung bình'
END
from SINHVIEN
```

Tổng kết

- Các câu lệnh trong SQL
- Các kiểu biến
- Các cấu trúc điều khiển

BÀI 05

STORED PROCEDURE VÀ TRIGGER TRONG SQL SERVER

NỘI DUNG BÀI 5

STORED PROCEDURE VÀ TRIGGER TRONG SQL SERVER

- Stored procedure
- Trigger

5.1 Stored Procedure Thủ tục thường trú

- Một Stored procedure bao gồm những thành phần chính sau:
 - Tên của stored procedure
 - Các tham số
 - Thân của stored procedure: bao gồm các lệnh của Transact-SQL dùng để thực thi procedure.
- Stored Procedure là hàm cho phép truyền tham số vào và trả về giá trị.
- Bao gồm 1 tập các lệnh T-SQL để xử lý 1 chức năng nào đó trong cơ sở dữ liệu
- Được tạo bằng lệnh Create Procedure, và có thể thay đổi bằng lệnh Alter Procedure, và có thể xóa bằng lệnh Drop Procedure trong lập lệnh của T_SQL.

sql.com

5.1 Stored Procedure Ưu điểm

- Cho phép lập trình theo Module
- Cho phép điều chỉnh chương trình cho phù hợp
- Cho phép thực thi nhanh và hiệu quả
- Giảm thiểu sự lộn xộn trên mạng
- Bảo mật và tùy chỉnh dễ dàng hơn

sql.com

5.1 Stored Procedure Phân loại

- System Stored Procedure:** Thủ tục hệ thống
- Local Stored Procedure:** Thủ tục người dùng
- Temporary Stored Procedure:** Thủ tục người dùng tạm
- Extended Stored Procedure:** Thủ tục chương trình ngoại vi đã biên dịch thành DLL
- Remote Stored Procedure:** Thủ tục sử dụng thủ tục của một server khác

sql.com

5.1 Stored Procedure Phân loại

Tên	Chức năng
sp_Databases	Đánh sách những Database có thể (available) trên Server (Danh sách này sẽ là khác nhau tùy thuộc vào quyền của người sử dụng)
sp_server_info	Chỉ tất cả những thông tin về Server, ví dụ như tập các đặc tính, phiên bản...
sp_stored_procedures	Đánh sách các tên thủ tục và thủ tục mỗi trường hiện tại
sp_tables	Đánh sách tất cả các bảng có thể trên mỗi trường hiện tại
sp_start_job	Khởi động tất cả các automated task ngay lập tức
sp_stop_job	Ngừng lại tất cả các automated task đang chạy
sp_password	Thay đổi password cho login account
sp_configure	Thay đổi lựa chọn cấu hình chung của SQL SERVER, khi người sử dụng không lựa chọn thì hệ thống sẽ hiển thị cấu hình mặc định.
sp_help	Hiển thị thông tin về bất kỳ đối tượng nào trong Database
sp_helptext	Hiển thị nội dung (text) của các đối tượng

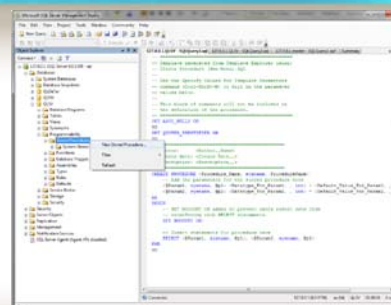
sql.com

5.1 Stored Procedure Tạo thủ tục

- Tạo bằng câu lệnh
- Tạo bằng SSMS

sql.com

5.1 Stored Procedure Tạo thủ tục bằng SSMS



sql.com

5.1 Stored Procedure Tạo thủ tục bằng câu lệnh

```
CREATE PROCEDURE procedure_name
    @parameter1 data_type [output] /*các tham số*/,
    @parameter2 data_type [output]
AS
BEGIN
    [khai báo các biến cho xử lý]
    (Các câu lệnh transact-sql)
END
GO

CREATE PROCEDURE XinChao
    @hoTen nvarchar(50)
AS
BEGIN
    print N'Xin chào ' + @hoTen
END
GO

CREATE PROC Hello
AS
BEGIN
    print N'Hello ' + @hoTen
END
GO
```

5.1 Stored Procedure Các dạng tham số

- Stored Procedure là 1 hàm được lưu trữ sẵn trong cơ sở dữ liệu. Hàm này có thể có 2 loại tham số chính
 - Tham số đầu vào : Đây là loại tham số mặc định, cho phép truyền các giá trị vào trong stored procedure để hỗ trợ xử lý.
 - Tham số đầu ra: dùng để nhận kết quả trả về từ stored procedure. Sử dụng từ khóa OUTPUT (hoặc viết tắt là OUT) để xác định tham số

5.1 Stored Procedure Các dạng tham số vào

```
127.0.0.1\QLSV - SQLQuery1.sql | Summary
CREATE PROC Cong
    @So1 int,
    @So2 int
AS
declare @Kq int
set @Kq = @So1 + @So2
print @Kq
GO
exec Cong 1, 2
```

5.1 Stored Procedure Các dạng tham số ra

```
--Tạo stored-procedure sp_tong
CREATE PROCEDURE sp_Tong
    @So1 int, @So2 int, @Tong int out
AS
Begin
    SET @Tong = @So1 + @So2;
End

--Biên dịch stored-procedure → F5

--Kiểm tra
Declare @Sum int
Exec sp_Tong 1, -2, out @Sum
Select @Sum
```

5.1 Stored Procedure Thực thi thủ tục

```
EXECUTE procedure_name parameter_value1, parameter_value2,...
EXEC procedure_name parameter_value1, parameter_value2, ...

127.0.0.1\QLSV - SQLQuery1.sql | Summary
CREATE PROCEDURE XinChao
    @hoTen nvarchar(50)
AS
Begin
    print N'Xin chào ' + @hoTen
End
EXEC XinChao N'Hiệp'
```

Xin chào Hiệp

Query executed successfully.

5.2 Trigger Khái niệm

- Trigger là một thủ tục đặc biệt mà việc thực thi của nó tự động khi có sự kiện xảy ra
- Không thể gọi trực tiếp Trigger và không nhận tham số
- Phân loại
 - DML triggers,
 - DDL triggers,
 - Logon triggers.

5.2 Trigger DML trigger

- Khái niệm:
 - DML triggers là các triggers sẽ được tự động thực thi khi các có sự kiện thao tác dữ liệu xảy tương ứng liên quan thao tác dữ liệu trên table hoặc view mà nó gắn kết.
 - DML là viết tắt của **data manipulation language**. Các sự kiện DML bao gồm các lệnh INSERT, UPDATE, hoặc DELETE thao tác trên các đối tượng table hoặc view.

5.2 Trigger DML trigger

- Mục đích
 - Dùng để kiểm soát việc thay đổi của dữ liệu
 - Luôn gắn với một Table cụ thể trong CSDL
 - Hoạt động khi có sự cập nhật (Insert, Update, Delete) dữ liệu trong Table
 - Đảm bảo toàn vẹn dữ liệu
 - Phát hiện, ngăn chặn hoặc hủy bỏ những thao tác thay đổi dữ liệu trái phép trong CSDL
 - Tạo và kiểm tra các ràng buộc giữa các bảng trong CSDL

5.2 Trigger Bảng logic trong DML trigger

- SQL Server định nghĩa hai bảng logic INSERTED và DELETED để sử dụng trong các Trigger
 - Cấu trúc 2 bảng giống cấu trúc của bảng mà Trigger tác động
 - Dữ liệu tùy thuộc ràng buộc và điều kiện kích hoạt Trigger
 - Lệnh Delete thực hiện thì các dòng bị xóa được sao chép vào DELETED (INSERTED rỗng)
 - Lệnh Insert thực hiện thì dòng mới được đưa vào INSERTED (DELETED rỗng)
 - Lệnh Update thực hiện thì các dòng xóa sẽ vào DELETED còn dòng chèn mới sẽ vào INSERTED

5.2 Trigger Tạo DML trigger

- Có hai loại DML trigger: **AFTER triggers** và **INSTEAD OF triggers**
 - Có thể ứng dụng DML trigger thay thế cho công việc của constraint nếu thấy thật cần thiết.
 - Một câu lệnh DML có thể sẽ tác động đến nhiều dòng trên table, tuy nhiên, chỉ có duy nhất một kích hoạt trigger xảy ra.

5.2 Trigger Tạo DML trigger

- Tạo và thực thi AFTER triggers
 - Có 3 dạng INSERT trigger, UPDATE Trigger và DELETE Trigger
 - Hoạt động của INSERT trigger
 - Bước 1: Câu lệnh insert được thực thi
 - Bước 2: Câu lệnh insert được ghi sổ
 - Bước 3: Các câu lệnh của After insert được thực thi
 - Hoạt động của UPDATE trigger
 - Bước 1: Câu lệnh update được thực thi
 - Bước 2: Câu lệnh update được ghi sổ
 - Bước 3: Các câu lệnh của After update được thực thi
 - Hoạt động của DELETE trigger
 - Bước 1: Câu lệnh delete được thực thi
 - Bước 2: Câu lệnh delete được ghi sổ
 - Bước 3: Các câu lệnh của After delete được thực thi

5.2 Trigger Tạo DML trigger

- Hoạt động của INSTEAD OF triggers
 - Bước 1: Câu lệnh **UPDATE, INSERT, hoặc DELETE** được thực thi
 - Bước 2: Câu lệnh được thực thi không xảy ra
 - Bước 3: Các câu lệnh của **INSTEAD OF trigger** được thực thi
- Dựa trên cách thức hoạt động của INSTEAD OF có thể thấy thực chất các câu lệnh insert, update, delete tác động trên table có gắn instead of trigger sẽ chỉ mang tính phát sinh sự kiện kích hoạt cho các lệnh bên trong trigger có thể thực thi.

5.2 Trigger

Ví dụ tạo DML trigger

```

Create Trigger trigger_name on table_name
For
As
Begin
    (Khái báo các biến xử lý)
    (Các lệnh Transact-SQL)
End

```

```

CREATE TRIGGER tg_delete ON human
FOR delete
AS
Begin
    DECLARE @idn int, @ma_contract smallint
    SELECT @idn = idn, @ma_contract = ma_contract
    FROM deleted
    UPDATE contract
    SET status = use
    WHERE idn = @idn AND ma_contract = @ma_contract
End

```

5.2 Trigger

DDL trigger

- DDL triggers là các trigger sẽ được tự động thực thi khi xảy ra các sự kiện liên quan đến định nghĩa dữ liệu trên phạm vi server hoặc database.
- DDL là viết tắt của **data definition language**.

5.2 Trigger

logon trigger

- Logon triggers là trigger được kích hoạt tương ứng với sự kiện LOGON.
- Sự kiện này được phát sinh khi user session được thiết lập đối với một instance của SQL Server.

Tổng kết

- Khái niệm Procedured, Trigger
- Phân loại
- Các loại tham số
- Các bước thực hiện

BÀI 06

QUẢN TRỊ SERVER

NỘI DUNG BÀI 6

QUẢN TRỊ SERVER

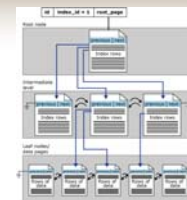
- Dùng chỉ mục (Index)
- Phân đoạn bảng
- Kiểm soát kích thước tập tin
- Kiểm tra tính toàn vẹn của dữ liệu
- Đặt giới hạn sử dụng tài nguyên hệ thống

6.1 Index

- Sử dụng như chỉ mục hay mục lục của mỗi quyển sách
- Tạo đường dẫn ngắn đến dữ liệu cần tìm
- Có cấu trúc dữ liệu dạng B-Tree
 - Khoảng cách từ gốc đến mọi nút là tương đương nhau
 - Thời gian tìm kiếm là với một B-Tree với n nút có chiều cao không quá $\log_2(n+1)/2$ (i là độ sâu)

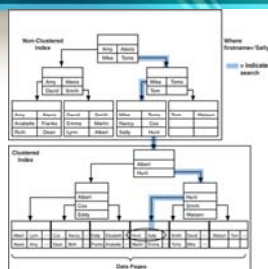
6.1 Index

- Index là công cụ rất mạnh để tăng hiệu năng của các câu lệnh tìm kiếm
- Có hai loại:
 - Clustered Index
 - NonClustered Index



6.1 Index Clustered Index

- Cách dùng
 - Sắp xếp bảng theo thứ tự của khóa Index
 - Toàn bộ bảng là 1 cây Index.
 - Tất cả các nút lá chứa khóa Index và đồng thời chứa tất cả các trường dữ liệu còn lại
- Lưu ý:
 - Mỗi bảng có tối đa một Clustered Index, có tên chứa 1 hoặc nhiều trường
 - Các bảng có Primary Key thì Clustered Index được tạo kèm theo



6.1 Index Non-Clustered Index

- Mỗi nút lá chứa khóa Index và con trỏ đến trang dữ liệu chứa bản ghi tương ứng
- Nếu bảng có Clustered Index thì con trỏ này chính là khóa Clustered Index, ngược lại là RowID
- Một bảng có thể có nhiều Non Clustered Index. Có thể chứa một hoặc nhiều trường
- Được lưu trữ tách rời khỏi bảng

Nonclustered Index



6.1 Index Clustered Index – Non Clustered Index

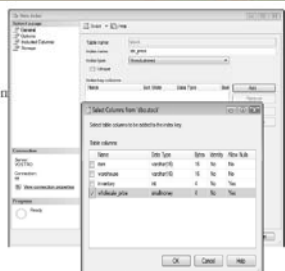
- Clustered Index :
 - Tránh bookmark lookup
 - Nâng cao độ ổn định cho Non Clustered Index
 - Chỉ được phép tạo một Clustered Index
- Non – Clustered Index
 - Có bookmark lookup nên hiệu suất thấp
 - Cho phép tạo nhiều Index trên bảng
 - Lưu trữ độc lập với bảng → Tăng hiệu suất xử lý song song

6.1 Index

- Mỗi Index có thể Unique hoặc Non Unique
- Index seek: Hệ thống có thể nhảy ngay đến nút cần tìm
- Index scan: Hệ thống quét cả cây Index để tìm

6.1 Index Tạo Index

- Tạo bảng công cụ
- Tạo bảng câu lệnh:
 - <http://msdn.microsoft.com/en-us/library/ms188783.aspx>



Ví dụ

- Dùng bảng Sales.Customer để tạo ra hai bảng mới:
 - Sales.Customer_noIndex và
 - Sales.Customer_Index
 - Tạo 1 Index trên trường CustomerID cho bảng Sales.Customer_Index

```
SELECT *
INTO Sales.Customer_NoIndex
FROM Sales.Customer
SELECT *
INTO Sales.Customer_Index
FROM Sales.Customer
GO
CREATE INDEX Idx_Customer_Index_CustomerID ON
Sales.Customer_Index(CustomerID)
```

Chọn cột đánh Index

- Cột là ứng cử viên tốt cho Index khi
 - Được sử dụng thường xuyên trong các điều kiện tìm kiếm
 - Được sử dụng trong điều kiện JOIN hai bảng
 - Độ lựa chọn đủ cao
- Ưu tiên Clustered Index cho cột
 - Tăng tuần tự
 - Kích thước không quá lớn
 - Được tìm kiếm với tần suất cao
 - Thường được tìm kiếm theo dải giá trị

Độ lựa chọn

Độ lựa chọn

Độ lựa chọn = Số giá trị khác biệt / Số bản ghi

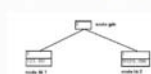
Độ lựa chọn cao

Cây index trên trường số CMT



Độ lựa chọn thấp

Cây index trên trường giới tính



Sử dụng Index

- Độ lựa chọn đủ cao
- Tránh chuyển đổi kiểu dữ liệu
- Tránh áp dụng hàm lên cột Index
- Cột đầu tiên trong khóa Index phải được sử dụng cho tìm kiếm

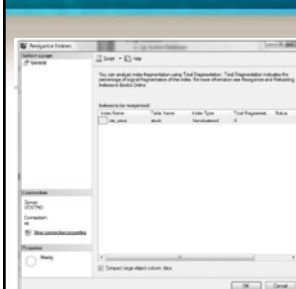
Bảo trì Index

- Các thao cập nhật làm Index bị phân mảnh
- Hai dạng phân mảnh
 - Hai nút kế tiếp không được lưu trữ liên tiếp nhau
 - Trang (page) chứa nhiều không gian trống
- Phân mảnh làm tăng số trang cần đọc cho cùng lượng dữ liệu => Giảm hiệu năng truy vấn

Bảo trì Index

- Rebuild và Reorganize
 - Dùng để cấu trúc lại Index, do đó giảm thiểu được phân mảnh
 - Re-Org thao tác nhanh hơn nhưng không hiệu quả khi Index bị phân mảnh nặng
 - Khi độ phân mảnh $\leq 30\%$ $\Rightarrow$ REORGANIZE
 - Khi độ phân mảnh $> 30\%$ $\Rightarrow$ REBUILD

Cập nhật Statistic



- Statistic chứa thông tin về phân bố dữ liệu của cột giúp Optimizer chọn phương án thực thi thích hợp
- Sau quá trình cập nhật dữ liệu, statistic bị quá hạn (outdate) có thể dẫn đến Optimizer chọn phương án sai

Phân đoạn bảng

- Table Partitioning
 - Kỹ thuật phân chia bảng thành từng đoạn nhằm quản lý CSDL có dung lượng lớn hiệu quả hơn
 - Đối với các ứng dụng truy cập từ bên ngoài, table vẫn duy nhất nhưng cấu trúc vật lý sẽ thay đổi

Lợi ích của Phân đoạn bảng

- Tiện lợi về quản trị
 - Backup/Restore
 - Rebuild lại Index
 - Hiệu quả với Switch – out và Switch – in
- Cải tiến hiệu năng
 - Chỉ cần truy cập vào đoạn cần thiết
 - Giảm tranh chấp I/O

Partition key

- Ví dụ:
 - Cho lược đồ: MatHang (MSMH, TenMH, DonGia, SoTon)
 - Phân đoạn theo đơn giá của mặt hàng
 - Giá các mặt hàng < 2500
 - Giá các mặt hàng $= 2500$
 - Giá các mặt hàng > 2500

Partition

- Partition Function
 - Create partition function partition_function_name (input_parameter_type) As Range [Left|Right] for values ([boundary_value [...,...]])
- Partition Scheme
 - Create partition scheme partition_scheme_name as Partition partition_function_name [ALL|TO] ([file_group_names] [Primary])
- Partition Table
 - Thêm lệnh: On {partition_scheme_name (partition_column_name)} sau câu lệnh Create Table

Bước 1: Tạo Part Func

- CREATE PARTITION FUNCTION PartFunc_2(DATETIME)
AS RANGE RIGHT FOR VALUES ('2010-01-01', '2011-01-01')

Partition	Partition 1	Partition 2	Partition 3
Dải giá trị	giá trị < 01/01/2010	01/01/2010 <= giá trị < 01/01/2011	01/01/2011 <= giá trị

Bước 2: Tạo Part Scheme

- CREATE PARTITION SCHEME MyPartScheme_1 AS
PARTITION MyPartFunc_1 TO (FG1, FG2, FG3, FG4)

Partition	Partition 1	Partition 2	Partition 3	Partition 4
Dải giá trị	giá trị <= 1000	1000 < giá trị <= 2000	2000 < giá trị <= 3000	3000 < giá trị
Filegroup	FG1	FG2	FG3	FG4

Bước 3: Tạo File Group

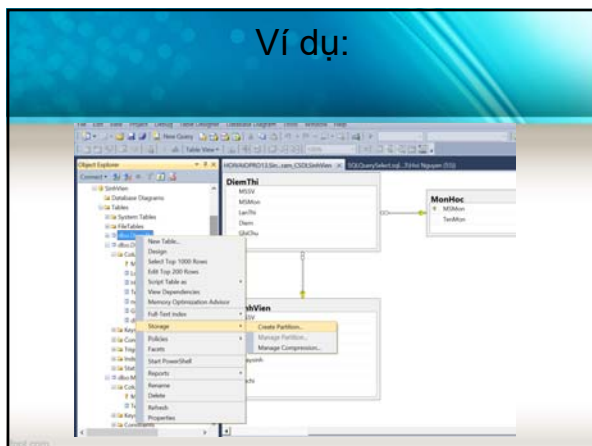
- ALTER DATABASE PartDB ADD FILEGROUP FG1
- ALTER DATABASE PartDB ADD FILEGROUP FG2
- ALTER DATABASE PartDB ADD FILEGROUP FG3
- ALTER DATABASE PartDB ADD FILEGROUP FG4
- ALTER DATABASE PartDB ADD FILE (NAME = N'F1', FILENAME = N'D:\DATA\PartDB_F1.ndf') TO FILEGROUP FG1
- ALTER DATABASE PartDB ADD FILE (NAME = N'F2', FILENAME = N'D:\DATA\PartDB_F2.ndf') TO FILEGROUP FG2
- ALTER DATABASE PartDB ADD FILE (NAME = N'F3', FILENAME = N'D:\DATA\PartDB_F3.ndf') TO FILEGROUP FG3
- ALTER DATABASE PartDB ADD FILE (NAME = N'F4', FILENAME = N'D:\DATA\PartDB_F4.ndf') TO FILEGROUP FG4

Bước 4: Tạo Part Table

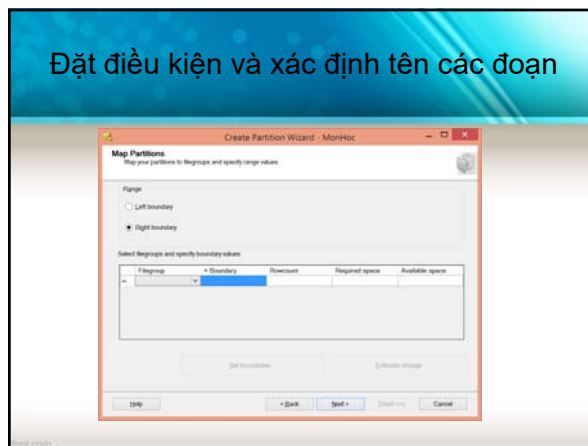
- CREATE TABLE dbo.TablePart(PartId INT
IDENTITY PRIMARY KEY, PartData
VARCHAR(100)) ON PartScheme_1(PartId)

Việc phân đoạn được gói gọn ở mệnh đề "ON PartScheme_1(PartId)", và bảng được phân đoạn thông qua partition scheme PartScheme_1 dựa vào trường PartId. Khi đó trường PartId được gọi là partition key

Ví dụ:



Đặt điều kiện và xác định tên các đoạn



Tổng kết

- Khái niệm Index và cách dùng
- Cách thức phân đoạn bảng và ý nghĩa
- Vì sao phải kiểm soát kích thước tập tin
- Toàn vẹn dữ liệu và kiểm soát
- Vì sao nên giới hạn tài nguyên hệ thống

BÀI 07 TỰ ĐỘNG HÓA QUẢN TRỊ

NỘI DUNG BÀI 7

TỰ ĐỘNG HÓA QUẢN TRỊ

- SQL Server Agent và tự động hóa quản trị
- Các đối tượng tham gia

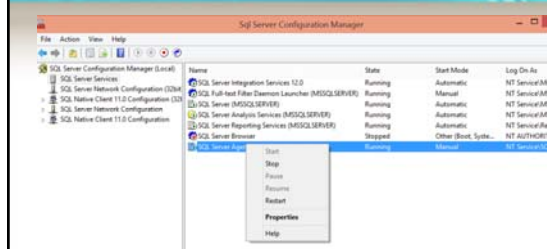
7.1 Tự động hóa quản trị

- Tự động hóa quản trị có vai trò quan trọng trong hoạt động quản trị CSDL của DN
- Thực hiện bằng việc thiết lập những tác vụ (Task) theo các lịch trình cụ thể
- SQL Server cho phép thiết lập tự động hóa hoạt động bằng SQL Server Agent

Quy trình thực hiện

- Khởi động SQL Server Agent
- Chọn Job
- Tạo Schedule
- Lựa chọn Alert
- Lựa chọn Operator
- Quản lý Error Logs

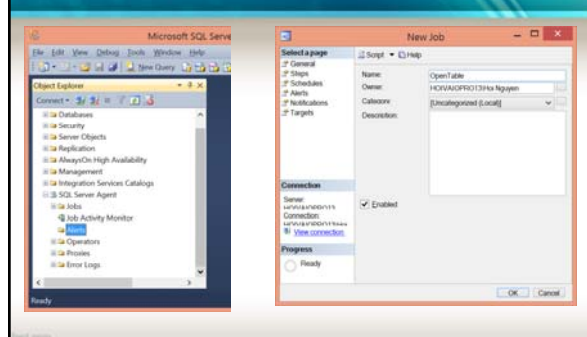
Khởi động SQL Server Agent



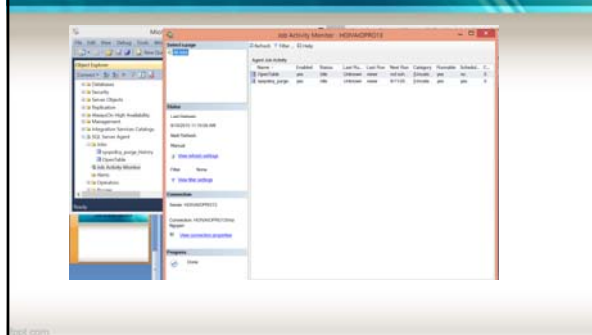
Chọn Job

- Job là một tập các Task mà SQL Server Agent sẽ thực hiện một hay nhiều lần có kết quả thành công hay thất bại
- Mỗi Job có nhiều bước (Step) mỗi bước tương ứng 1 hành động
- Có thể tạo bằng Tools hoặc T-SQL
- Thực thi Job bằng lập lịch (Schedule) → Dựa vào các Alert để tiếp tục thực thi → Sử dụng hàm lưu trữ hệ thống `sp_start_job`
- Job có thể thực thi trên PC hoặc WorkStation hoặc Remote Control

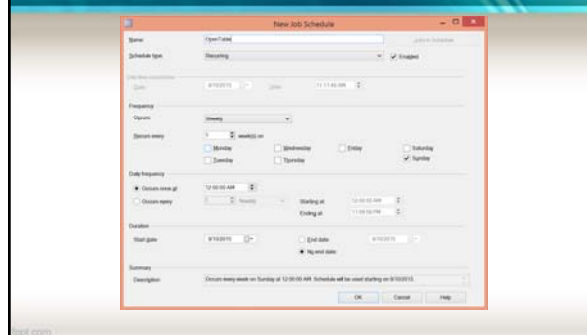
Job properties



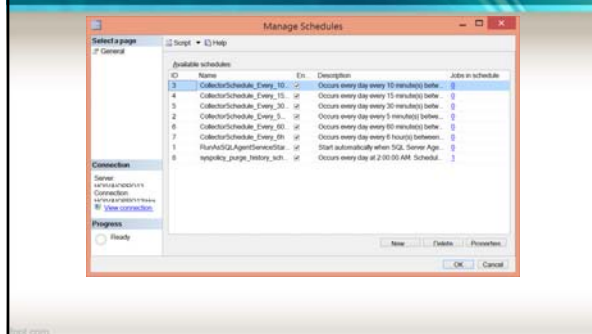
Job Activity Monitor



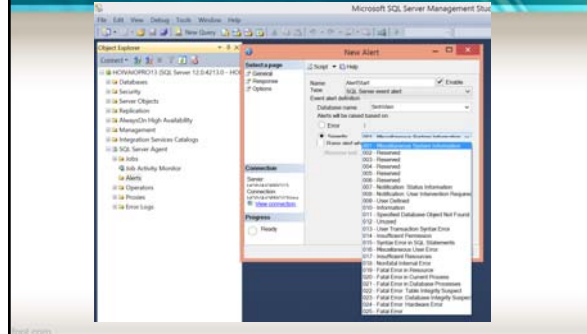
Schedule



Manager Schedule



Alert (Thông báo khi thực thi)



Đặt lịch

- Name: Đặt tên lịch trình
- Schedule type: kiểu lịch trình
- Frequency: Tần suất thực hiện
- Duration: Thời gian thực hiện

B3: Khai báo Alert

Tên Thông báo
Kiểu thông báo
CSDL sẽ được
thông báo
Nội dung dòng
thông báo

B4: Khai báo Operator

Thực thi Job

Thực thi Job
Kích chuột phải chọn
Start Jobs
Xem thực thi và đợi
cảnh báo

Kết thúc

- Tìm hiểu về SQL Server Agent
- Thực hiện ví dụ
- Kiểm tra và đọc các lỗi

BÀI 08 ĐỒNG BỘ HÓA CSDL

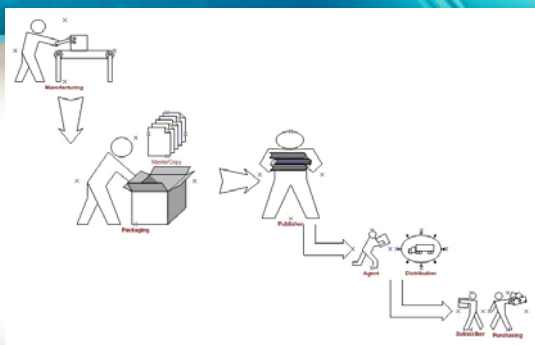
Nội dung bài 08

- Cơ chế và các kiểu đồng bộ hóa
- Các bước đồng bộ hóa dữ liệu
- Mô hình đồng bộ hóa dữ liệu
- Cài đặt quy trình đồng bộ hóa dữ liệu

8.1 Cơ chế và các kiểu đồng bộ

- Đồng bộ dữ liệu
 - Đồng bộ (Replication) là kỹ thuật dùng để sao chép và phân phối dữ liệu và các đối tượng từ một CSDL này đến một CSDL khác và cho phép cập nhật các thay đổi trên CSDL đảm bảo tính nhất quán của CSDL

Cơ chế



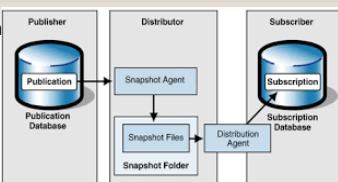
Các kiểu đồng bộ hóa

- Có 3 kiểu chính:
 - Snapshot replication
 - Transactional replication
 - Merge replication

Snapshot Replication

Cơ chế:

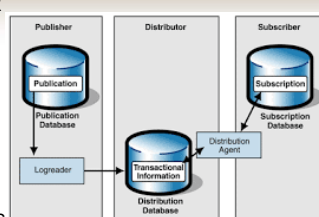
- Các đối tượng CSDL chuyển từ Publisher đến Subscriber sẽ được ghi đè lên nếu chúng đang tồn tại trong CSDL của Subscriber
- Phù hợp nhất với CSDL tĩnh, ít biến động
- Yêu cầu ràng buộc kết nối giữa hai CSDL khi thực hiện



Transactional Replication

Cơ chế:

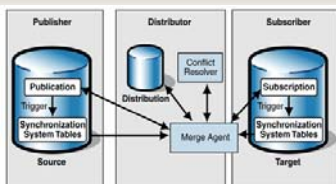
- Tạo bản sao cho mỗi task đã được định nghĩa trong đối tượng CSDL của Publisher
- Phù hợp với CSDL nhiều biến động do đồng bộ dữ liệu nhanh, cập nhật được gần với thời thực
- Yêu cầu ràng buộc kết nối giữa hai CSDL khi thực hiện



Merge Replication

Cơ chế:

- Được dùng khi không có bất kỳ ràng buộc kết nối giữa Publisher và Subscriber
- Tổng hợp dữ liệu thay đổi giữa Publisher và Subscriber trong lần kết nối tiếp theo
- Có khả năng tự động xử lý tranh chấp dữ liệu khi tổng hợp



8.2 Thành phần đồng bộ hóa

- Có 3 thành phần cần xem xét:
 - Đối tượng đồng bộ dữ liệu
 - Dữ liệu đồng bộ hóa
 - Các dịch vụ

Nhóm đối tượng đồng bộ

- Publisher:
 - Máy chủ CSDL tập trung tài nguyên để truyền gửi/phát hành
- Subscriber:
 - Máy chủ CSDL nơi nhận dữ liệu
 - Có thể xem là Publisher khi chuyển dữ liệu đến Subscriber khác
- Distributor:
 - Cầu nối trung gian giữa Publisher và Subscriber
 - Có thể xem là CSDL đặt trên máy chủ Publisher hoặc Subscriber

Dữ liệu đồng bộ hóa

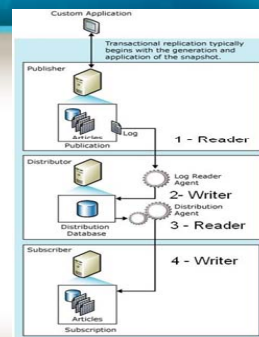
- Article:
 - Tập dữ liệu được cấu hình để đồng bộ hóa
 - Các đối tượng CSDL như Table, View, Sp, Data và các ràng buộc trong Table
- Publication:
 - Gồm nhiều Article được xuất bản cho Subscriber
 - Cho phép tạo bản sao cho nhóm gồm nhiều Article
- Subscription:
 - Được xem như đơn đặt hàng để nhận Article từ nhiều Publication, bao gồm cả các ràng buộc khác khi thực hiện phân phối Article

Các đối tượng khác

- SQL Server Agent:
 - Đóng vai trò chính trong kiểm soát, vận hành tiến trình đồng bộ hóa và thực thi theo thời gian thực hoặc lịch đã lập sẵn
- Snapshot Agent
 - Nhận và thực thi snapshot cho Snapshot replication, Transactional replication, Merge replication
- Log Reader Agent
 - Đọc bản ghi Task trong Publisher và ghi lại các bản ghi này cho mỗi Article được xuất bản cho Distributor
- Distributor Agent
 - Đọc task đã ghi vào CSDL Distributor và áp dụng cho Subscriber
- Merge Agent
 - Quản lý hoạt động của Merge replication

Hoạt động cơ bản

- Cơ chế:
- 1 Reader
 - 2 Writer
 - 3 Reader
 - 4 Writer



8.3 Các mô hình đồng bộ hóa

- Một Publisher và một Subscriber
- Một Publisher và nhiều Subscriber
- Nhiều Publisher và một Subscriber
- Nhiều Publisher và nhiều Subscriber

Một Publisher và một Subscriber

• Một Publisher và một Subscriber

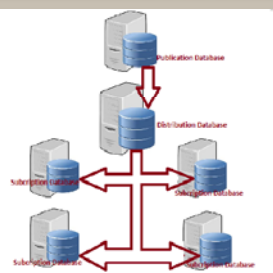
- Sử dụng khi phân phối dữ liệu từ CSDL trung tâm đến một CSDL trạm.
- Publication Database và Distribution Database có thể đặt trên hai server khác nhau.



Một Publisher và nhiều Subscriber

• Một Publisher và nhiều Subscriber

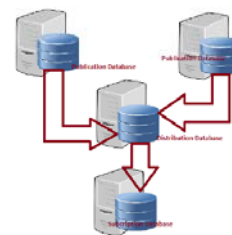
- Sử dụng khi phân phối dữ liệu từ CSDL trung tâm đến nhiều CSDL trạm.



Nhiều Publisher và một Subscriber

• Nhiều Publisher và một Subscriber

- Sử dụng khi lấy dữ liệu từ CSDL trạm về CSDL trung tâm



Nhiều Publisher và nhiều Subscriber

• Nhiều Publisher và nhiều Subscriber

- Sử dụng khi lấy dữ liệu từ CSDL trạm về CSDL trung tâm và ngược lại.



Các kiểu di chuyển dữ liệu

- Push Subscriber
 - Publisher đẩy (push) những cập nhật đến Subscriber mà không quan tâm Subscriber có cập nhật hay không
 - Dùng khi ứng dụng yêu cầu gửi các cập nhật đến cho Subscriber ngay khi có thay đổi ở Publisher
 - Bảo mật cao, các Subscriber thấp
- Pull Subscriber
 - Subscriber kéo những cập nhật tại Publisher về theo một khoảng thời gian định kỳ
 - Phù hợp cho các Subscriber có độc lập cập nhật
 - Khả năng bảo mật thấp
 - Cho phép số lượng Subscriber cao

8.4 Cài đặt đồng bộ hóa

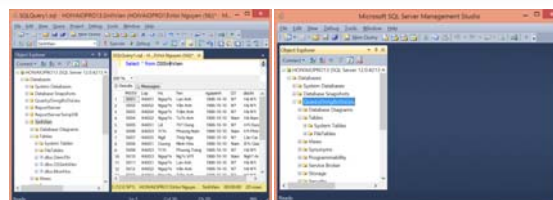
- Các bước
 - Tạo 1 CSDL không chứa bảng
 - Tạo một truy vấn đến dữ liệu
 - Tạo Publication
 - Cài đặt thông số cho Publication
 - Tạo Subscriber
 - Cài đặt thông số cho Subscriber
 - Kiểm tra

not.com

Các bước thực hiện

Publisher
SinhVien

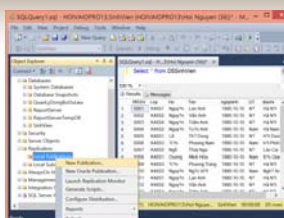
Subscriber
QuanLyDongBoDuLieu



not.com

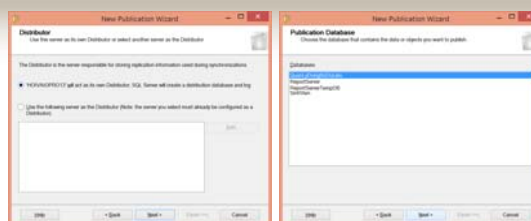
Tạo Publication

- Ví dụ tạo Snapshot Replication
 - Chọn Replication
 - Chọn Local Publication
 - Chọn New Publication



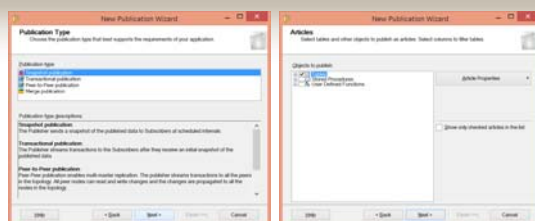
not.com

Cài đặt thông số



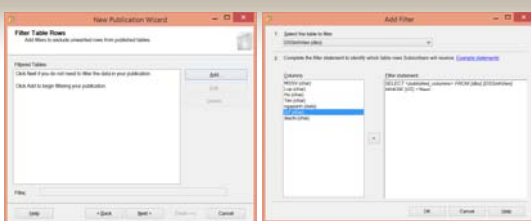
not.com

Cài đặt thông số



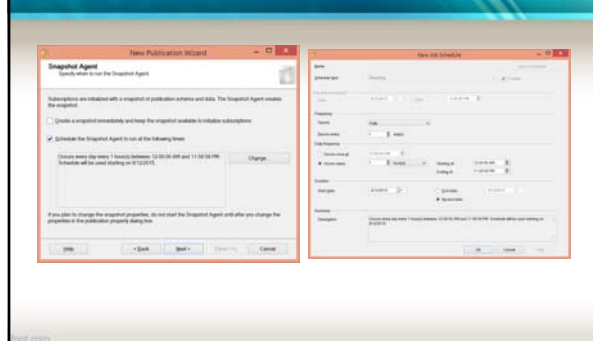
not.com

Cài đặt thông số

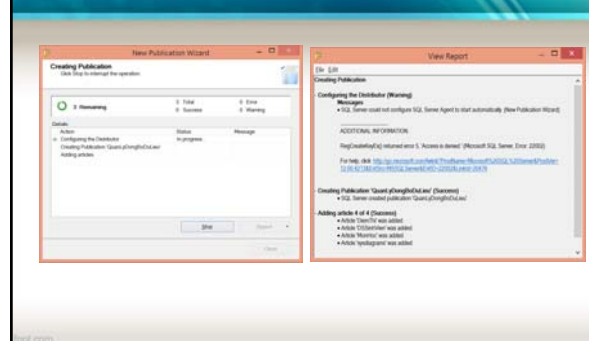


not.com

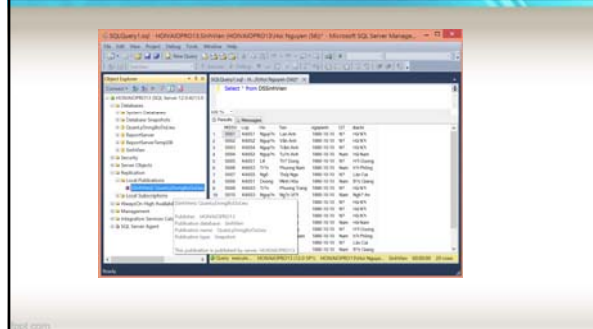
Chọn chế độ tức thời hay theo lịch



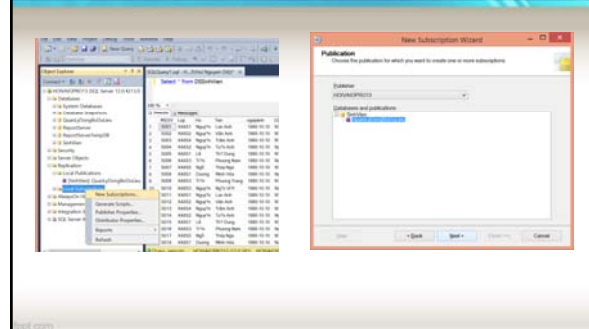
Thực thi và xem kết quả



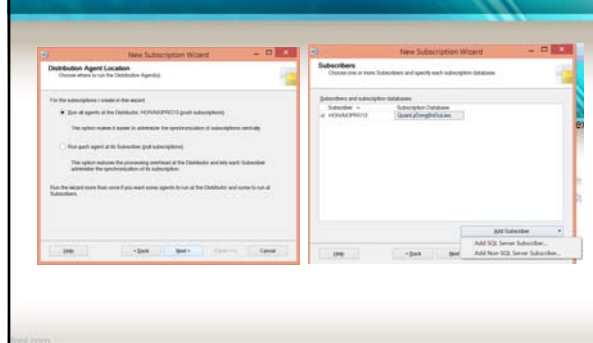
Sau khi tạo xong thông số Publication



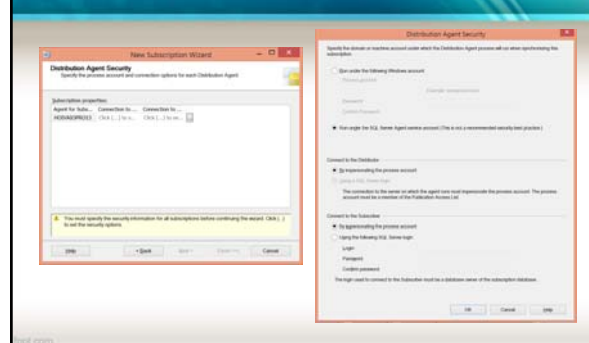
Tạo Subscriber Chọn các thông số



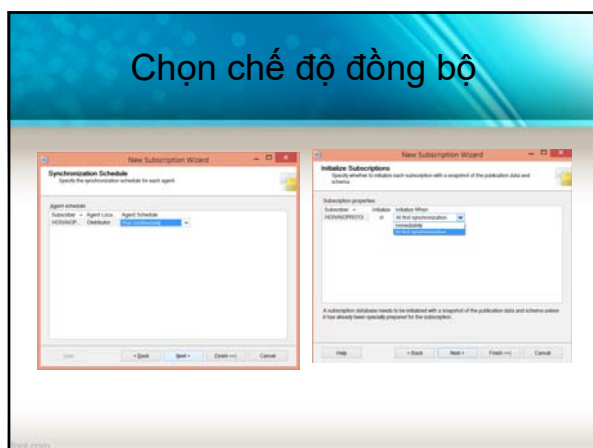
Chọn và cài đặt thông số



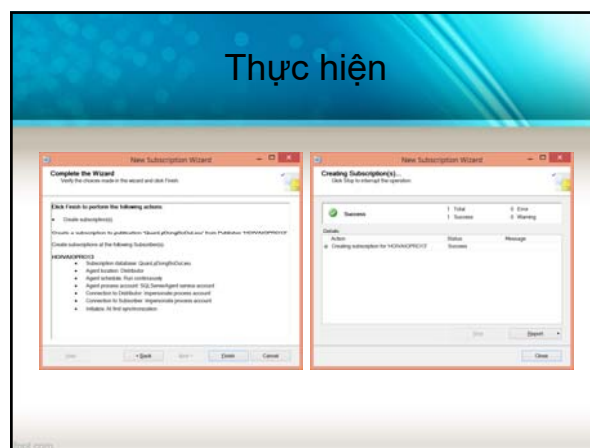
Cài đặt độ bảo mật



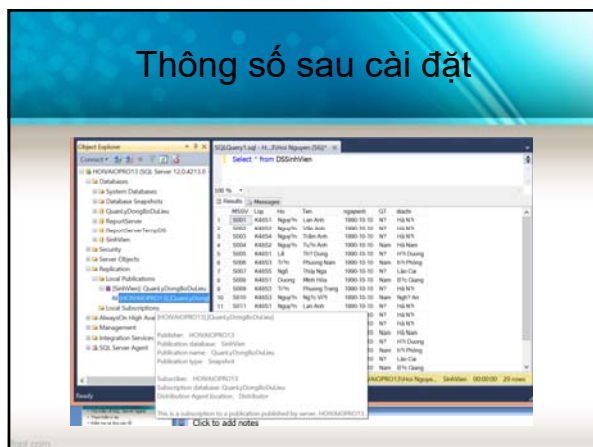
Chọn chế độ đồng bộ



Thực hiện



Thông số sau cài đặt



Kết thúc

- Tìm hiểu về SQL Server Agent
- Thực hiện ví dụ tạo Publication và Subscriber
- Kiểm tra và đọc các lỗi

BÀI 10 TOÀN VẬN DỮ LIỆU

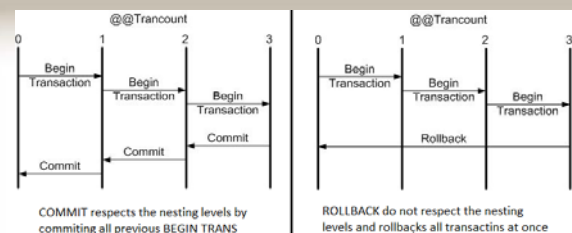
Nội dung bài 10

- Giao dịch và tính chất của giao dịch
- Sự bắt đầu và sự kết thúc một giao dịch
- Mức độ tách biệt/khác nhau của giao dịch
- Khóa

10.1 Giao dịch và tính chất của giao dịch

- Transaction là một đơn vị tác vụ bao gồm một tập có thứ tự các tác vụ con
 - Các tác vụ con được sắp xếp theo một trình tự xác định
 - Các tác vụ con hoàn thành thì Transaction hoàn thành
 - Nếu Transaction hoàn thành thì các dữ liệu biến động được cập nhật lên CSDL
 - Nếu Transaction không hoàn thành có thể Roll Back hoặc Cancel => dữ liệu không được cập nhật lên CSDL
 - Một Transaction có 4 tính chất: Atomicity, Consistency, Isolation and Durability (ACID)

Ví dụ về Transaction



Tính chất của Transaction

- Atomicity – Nguyên tử**
 - Mỗi Transaction là một đơn vị nhỏ nhất, các dữ liệu có thể được thao tác hoặc không
- Consistency – Nhất quán**
 - Transaction sẽ không thực hiện nếu có một thao tác có lỗi về mặt logic hoặc ràng buộc
- Isolation – Tách biệt/ Độc lập**
 - Tại mỗi thời điểm các Transaction được thực hiện đồng thời và chúng chỉ tác động với nhau khi dữ liệu được cập nhật hay kết thúc phiên
- Durability – Bền vững**
 - Đảm bảo sau khi Transaction thành công các thay đổi đã được tạo ra trên CSDL vẫn được duy trì

ACID trong SQL Server



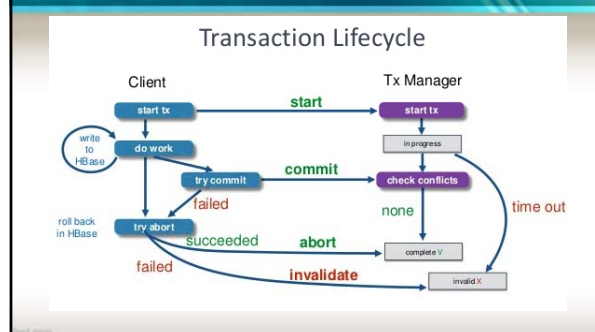
Kết thúc Transaction

- Kết thúc giao dịch có 2 kiểu
 - Xác nhận giao dịch kết thúc
 - Commit [Transaction] [Transaction_name] @Transaction_name_variable]
 - Hủy bỏ và quay lại giao dịch
 - Rollback [Transaction] [Transaction_name] @Transaction_name] Savepoint_name] @Savepoint_name]

10.2 Bắt đầu và kết thúc một Transaction

- Có 3 loại Transaction
 - Explicit Transaction: Phiên giao dịch rõ (tường minh)
 - BEGIN [Transaction] [Transaction_name] @Transaction_name_variable]
 - Implicit Transaction : Phiên giao dịch ẩn (không tường minh)
 - SET Implicit_Transaction {On|Off}
 - Autocommit Transaction: Mỗi câu lệnh tự cập nhật dữ liệu khi nó kết thúc, không cần câu lệnh điều khiển phiên giao dịch

Vòng đời của một Transaction



Ví dụ Explicit Transaction

```

-- It's basically the same logic than
-- our previous batch, with only a few
-- changes
declare @i bigint
declare @icnt int
select @icnt=1
-- Now we have a transaction during the
-- entire operation
begin transaction
while @icnt<3
begin
select @i=@i+1
while @i<1032192
begin
-- ...
end
select @icnt=@icnt+1
end
commit transaction
  
```

Ví dụ Implicit Transaction

```

use tempdb
GO
-- Create Test Table
CREATE TABLE TruncateTest (ID INT)
INSERT INTO TruncateTest (ID)
SELECT 1
UNION ALL
SELECT 2
UNION ALL
SELECT 3
GO
-- Check the data before truncate
SELECT * FROM TruncateTest
GO
-- Begin Transaction
BEGIN TRAN
-- Truncate Table
TRUNCATE TABLE TruncateTest
GO
-- Check the data after truncate
SELECT * FROM TruncateTest
GO
-- Rollback Transaction
ROLLBACK TRAN
GO
-- Check the data after Rollback
SELECT * FROM TruncateTest
GO
-- Clean up
DROP TABLE TruncateTest
GO
  
```

Ví dụ Autocommit Transaction

```

USE tempdb
GO
-- Create Test Table
CREATE TABLE TruncateTest (ID INT)
INSERT INTO TruncateTest (ID)
SELECT 1
UNION ALL
SELECT 2
UNION ALL
SELECT 3
GO
-- Check the data before truncate
SELECT * FROM TruncateTest
GO
-- Begin Transaction
BEGIN TRAN
-- Truncate Table
TRUNCATE TABLE TruncateTest
GO
-- Check the data after truncate
SELECT * FROM TruncateTest
GO
-- Rollback Transaction
ROLLBACK TRAN
GO
-- Check the data after Rollback
SELECT * FROM TruncateTest
GO
-- Clean up
DROP TABLE TruncateTest
GO
  
```

10.3 Mức độ tách biệt của Transaction

- SQL Server xác định 5 mức độ tách biệt của các Transaction
 - Read UnCommitted
 - Read Committed
 - RepeatTable Read
 - Serializable
 - Snapshot

Read UnCommitted

- Là mức độ thấp nhất, cho phép Transaction đọc dữ liệu được viết bởi Transaction khác chưa được xác nhận thay đổi. Nói cách khác có thể đọc dữ liệu gốc khi đang có Transaction sửa đổi dữ liệu.
- Thiết lập:
 - Set Transaction Isolation Level Read UnCommitted
- Nhược điểm
 - Có thể đọc dữ liệu không đúng từ một UnCommitted khác chưa RollBack (Dirty Reads)
 - Đọc dữ liệu "ma" từ kết quả Insert hoặc Delete của một Transaction khác (Phantom Reads)
 - Đọc dữ liệu không lặp lại từ kết quả cập nhật dữ liệu của một Transaction khác (NonRepeatable Reads)

Các lỗi hay gặp

- **Đọc dữ liệu sai (Dirty reads)**
 - Xảy ra khi giao dịch đọc một bản ghi mà một phần giao dịch khác chưa hoàn thành. Nếu giao dịch trước đó chưa hoàn thành hay đang thực hiện chế độ Rollback chúng ta sẽ phải đọc dữ liệu cũ, dữ liệu sai.
- **Đọc bản ghi hai lần (Unrepeatable reads)**
 - Khi đọc mẫu tin hai lần trong một giao dịch trong khi giao dịch khác chỉ thông báo về tình trạng dữ liệu trong một khoảng thời gian quy định
- **Phantoms (đọc các bản ghi không có)**
 - Nghĩa là chúng ta đọc được những bản ghi không có. Vì những bản ghi đó xuất hiện không bị tác động bởi các lệnh UPDATE hoặc DELETE.
- **Lost Update (cập nhật mất DL)**
 - Xảy ra khi một giao dịch cập nhật dữ liệu vào cơ sở dữ liệu thành công, nhưng lại ghi đè lên dữ liệu của giao dịch khác.
 - Xảy ra khi hai giao dịch đang đọc bản ghi dữ liệu, sau đó giao dịch 1 ghi dữ liệu của bản ghi, giao dịch 2 cũng ghi kết quả chỉ có giao dịch 2 được cập nhật

Read Committed

- Là mức độ được thiết lập mặc định trong SQL Server, cho phép Transaction có thể đọc dữ liệu gốc khi Transaction xác nhận thay đổi
- Thiết lập
 - Set Transaction Isolation Level Read Committed
- Nhược điểm
 - Gặp lỗi Phantom Reads và NonRepeatable

Repeatable Read

- Loại bỏ được NonRepeatable Reads bằng việc ngăn một Transaction không được thay đổi dữ liệu gốc khi có một Transaction khác đang đọc dữ liệu gốc cho đến khi Transaction này commit hoặc Rollback
- Thiết lập
 - Set Transaction Isolation Level Repeatable Read

Serializable

- Loại bỏ được Phantom Reads bằng việc ngăn một Transaction không được chèn hoặc xóa bản ghi dữ liệu gốc vào phạm vi truy cập (Tập dữ liệu) của một Transaction khác đang thực hiện đến khi Transaction này commit hoặc Rollback
- Thiết lập
 - Set Transaction Isolation Level Serializable

Snapshot

- Tạo một bản sao Snapshot của dữ liệu gốc khi Transaction bắt đầu thực thi và cho phép Transaction thực thi trên bản sao này cho đến lúc commit hoặc Rollback
- Thiết lập
 - Set Transaction Isolation Level Snapshot

So sánh các mức độ tách biệt

Isolation level	Dirty read	Nonrepeatable read	Phantom
Read uncommitted	Yes	Yes	Yes
Read committed	No	Yes	Yes
Read committed using row versioning	No	Yes	Yes
Repeatable read	No	No	Yes
Snapshot	No	No	No
Serializable	No	No	No

Khóa - Lock

- Locks là cơ cấu cho phép ngăn ngừa các hành động trên đối tượng có thể gây ra xung đột với những gì đã thực hiện và hoàn thành trên đối tượng trước đó.
- Khi làm việc trên cơ sở dữ liệu đa người dùng, xung đột giữa nhiều người sử dụng cùng thực hiện là thường xuyên xảy ra.
- Xử lý đụng độ hay tranh chấp trên đối tượng, chúng ta phải biết khi nào nên khoá (lock) khi nào không thể khoá, và những loại lock nào đang có.

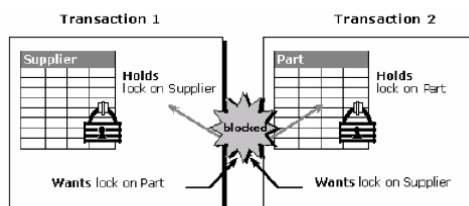
Lock

- Lock hướng đến giải quyết 4 vấn đề sau
 - Dirty reads (đọc dữ liệu sai)
 - Unrepeatable reads (đọc hai lần bản ghi)
 - Phantoms (Đọc các bản ghi nháp, không có)
 - Lost updates (cập nhật, mất dữ liệu)

Các loại Lock trong SQL Server

- Pessimistic Lock:
 - Là chiến lược lock trước tài nguyên (rows) trước khi end users thay đổi
- Optimistic Lock
 - Là chiến lược chỉ lock tại thời điểm user đang thay đổi dữ liệu.
- Shared Locks (S):
 - Hạn chế quyền sửa của người dùng 2 khi người dùng 1 đang đọc hoặc truy cập dữ liệu
- Exclusive Lock (X)
 - Hạn chế quyền đọc và sửa của người dùng 2 khi người dùng 1 đang cập nhật dữ liệu
- Update Lock (U)
 - Kết hợp giữa share lock và exclusive lock.

DeadLock



Phạm vi khóa

- Phạm vi khóa

Tên	Mô tả
Page	Trang dữ liệu 2K hoặc trang chỉ mục Index, thường được dùng
Extent	Nhóm các trang có kích thước 8k, chỉ dùng với trường hợp xác định
Table	Cả bảng dữ liệu, gồm dữ liệu và index
Intent	Là kiểu đặc biệt để đặt kiểu khoá của trang hiện tại trên bảng

	RL	ROW-R	ROW-S	ROW-M	S	U	X	IS	IX	SI	SIX	SI-X	BU	RS-S	RS-U	RS-M	RS-X	RS-S	RS-U	RS-M	RS-X	RS-S	RS-U	RS-M	RS-X	RS-S	RS-U	RS-M	RS-X
RL	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
ROW-R	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
ROW-S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
X	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
IS	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
IX	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
SI	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
SIX	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
BU	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-M	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-X	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-M	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-X	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-M	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
RS-X	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

N	No Conflict	SIU	Share with Intent Update
I	Illegal	SDX	Shared with Intent Exclusive
C	Conflict	UIX	Update with Intent Exclusive
RL	No Lock	BU	Bulk Update
ROW-S	Schema Stability Locks	RS-S	Shared Range-Shared
ROW-R	Schema Modification Locks	RS-U	Shared Range-Update
S	Shared	RS-M	Shared Range-Modify
U	Update	RS-X	Shared Range-Exclusive
X	Exclusive	RS-S	Insert Range-Shared
IS	Intent Shared	RS-U	Insert Range-Update
IU	Intent Update	RS-M	Insert Range-Modify
IX	Intent Exclusive	RS-X	Insert Range-Exclusive
		RS-S	Exclusive Range-Shared
		RS-U	Exclusive Range-Update
		RS-M	Exclusive Range-Modify
		RS-X	Exclusive Range-Exclusive

Kết thúc

- Tạo sao cần bảo mật CSDL?
- Các chế độ bảo mật CSDL của SQL server
- SQL Server có những người dùng nào? Quyền và vai trò của người dùng trong SQL Server
- Các chế độ mã hóa trong SQL Server

BÀI 11

SAO LƯU, PHỤC HỒI DỮ LIỆU

Nội dung bài 11

- SAO LƯU và PHỤC HỒI
- CÁC KIỂU SAO LƯU VÀ PHỤC HỒI
- SAO LƯU TỰ ĐỘNG

11.1 Sao lưu và phục hồi

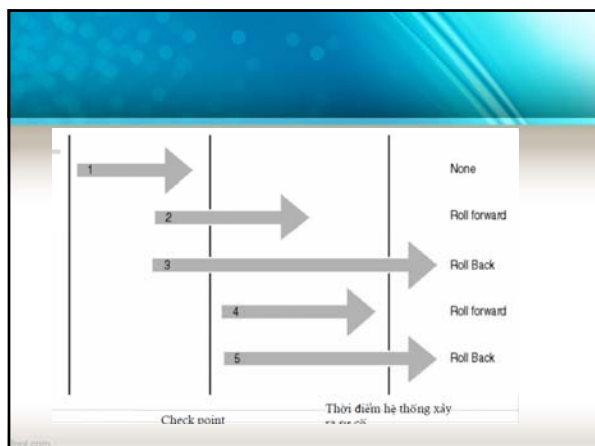
- Dữ liệu có thể bị mất mát, hư hỏng khi thực thi, lưu trữ, truyền
- => Để tránh mất mát, hư hỏng => Thường xuyên sao lưu => Khi hệ thống dữ liệu gặp các vấn đề có thể phục hồi từ các bản sao

Sao lưu

- Sao lưu CSDL là tạo thêm một bản sao CSDL
- Bản sao có thể dùng để khôi phục lại CSDL trong trường hợp CSDL gặp sự cố
- Bản sao thường gồm tất cả các file dữ liệu và file transaction log
- File log để có thể rollback hoặc roll forward các trạng thái giao dịch trước đó

Các trạng thái trong log

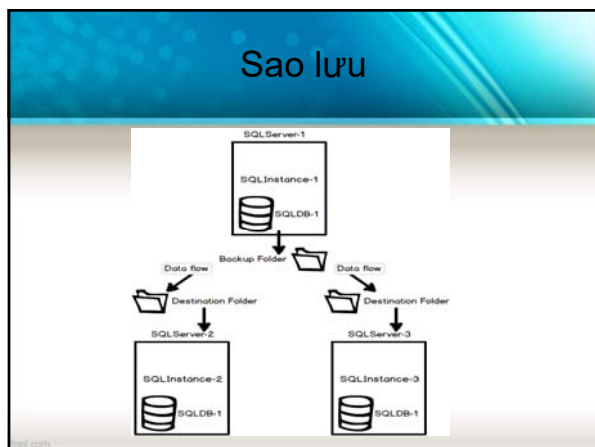
- **Roll forward** là khôi phục tất cả các giao dịch đã hoàn thành khi hệ thống xảy ra sự cố.
- **Roll back** là hủy bỏ giao dịch chưa hoàn thành khi hệ thống xảy ra sự cố.
- **Check point** là thời điểm ghi lại tất cả những trạng dữ liệu thay đổi trên đĩa.



Sao lưu

- Sao lưu một CSDL là ghi lại toàn bộ trạng thái của CSDL tại thời điểm thực thi
- Sao lưu Transaction Log
- Trong quá trình sao lưu SQL Server vẫn cho phép thực hiện các giao dịch

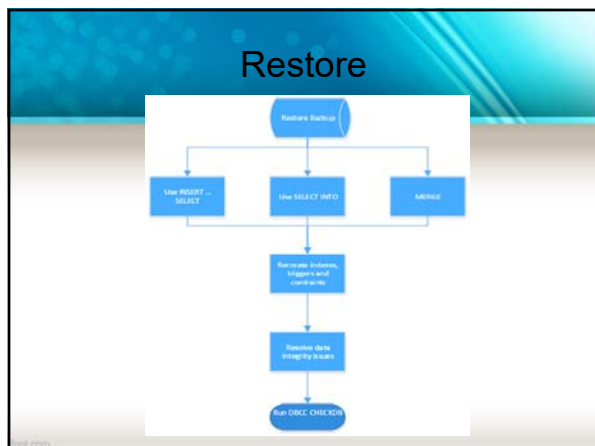
Sao lưu



Phục hồi

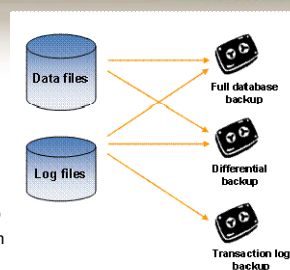
- Phục hồi là khôi phục một bản sao lưu CSDL để đưa về trạng thái khi sao lưu
- Các Transaction không hoàn thành trong trạng thái đó sẽ được Roll Back để đảm bảo tính nhất quán của CSDL
- Khôi phục một bản Transaction Log

Restore

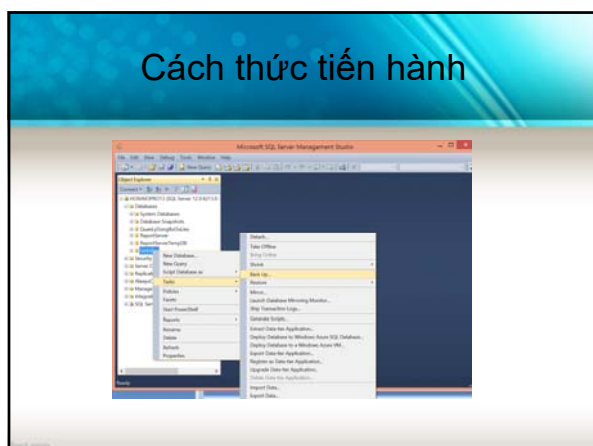


Các loại backup

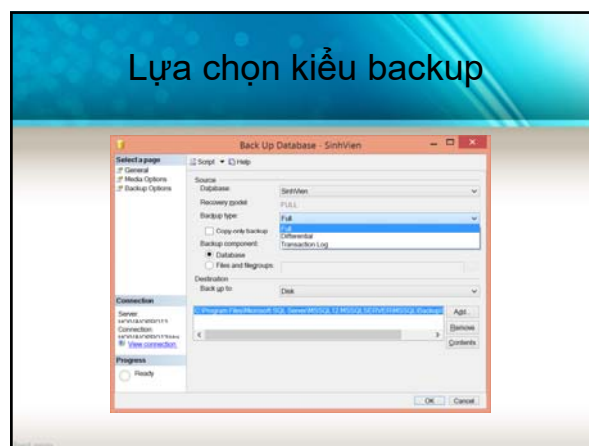
- Full Database Backup
 - Sao lưu toàn bộ
- Differential Backup
 - Chỉ sao lưu các thay đổi
- Transaction Log Backup
 - Chỉ sao lưu các Transaction có thay đổi trong lần gần nhất
- File or File Group Backup
 - Chỉ sao lưu File hoặc nhóm file



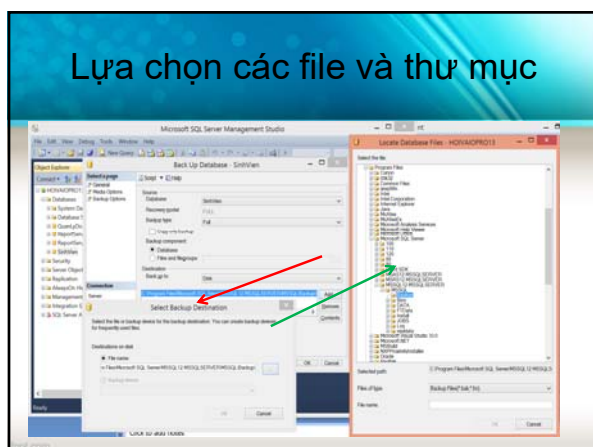
Cách thức tiến hành



Lựa chọn kiểu backup

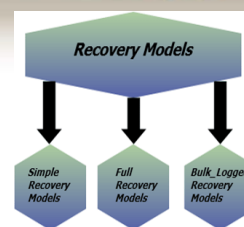


Lựa chọn các file và thư mục

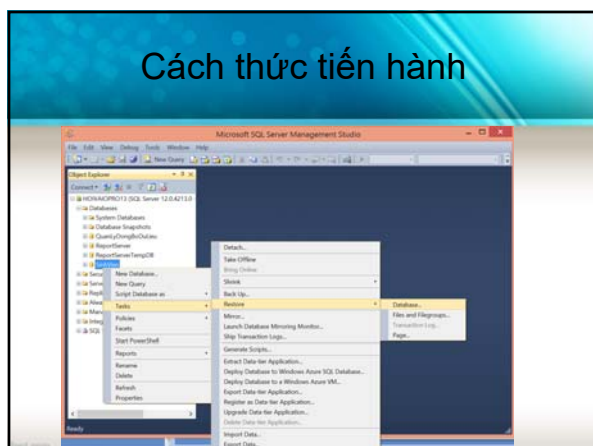


Phục hồi

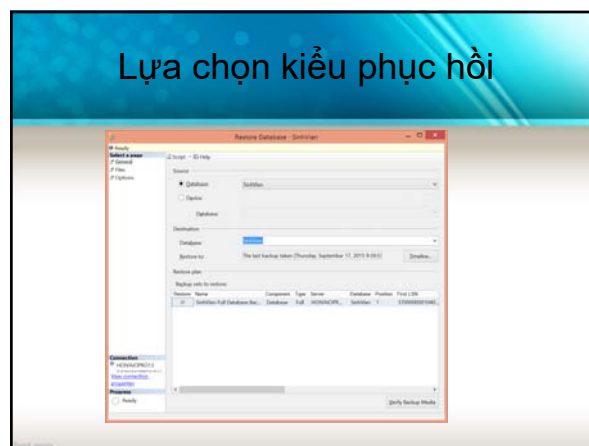
- Full Recovery Model
 - Có thể phục hồi gần toàn bộ về một thời điểm trong quá khứ
- Bulk Logged Recovery Model
 - Các hoạt động Log đầy đủ và Log minimum
- Simple Recovery Model
 - Chỉ phục hồi CSDL về thời điểm sao lưu gần nhất



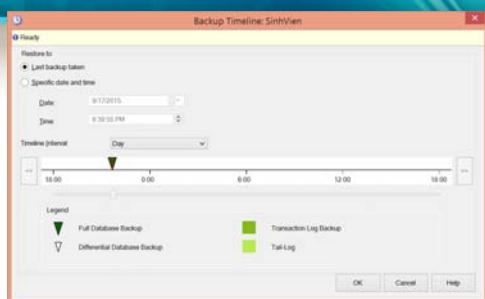
Cách thức tiến hành



Lựa chọn kiểu phục hồi



Chọn thời điểm cần phục hồi



Một số chú ý khi sao lưu và phục hồi

- Sao lưu
 - Thường xuyên sao lưu để giảm rủi ro cho hệ thống
 - Khi có bất kỳ thay đổi quan trọng nào trên CSDL nên sao lưu
 - Full Backup là giải pháp tối ưu nhưng tốn không gian nhớ

Một số chú ý khi sao lưu và phục hồi

- Phục hồi
 - Cần theo dõi các bản phục hồi thật chính xác
 - Lựa chọn các bản phục hồi theo các sự cố
 - Chú ý lựa chọn các kiểu phục hồi phù hợp theo tổn thất sự cố gây ra cho hệ thống

Sao lưu tự động

- Sử dụng Agent SQL Server để tạo lịch sao lưu tự động

Kết thúc

- Tạo sao cần sao lưu CSDL?
- Các chế độ sao lưu CSDL của SQL server
- SQL Server có những cơ chế phục hồi nào? Khi nào nên phục hồi một CSDL trong SQL Server
- Cài đặt chế độ sao lưu tự động trong SQL Server