

1. Điều kiện để phân tán được CSDL: có dịch vụ SQL Server Agent
2. Phân tán cơ sở dữ liệu có 3 bước:
 - Định nghĩa Server **Distributor** : chứa DB distribution
 - Định nghĩa publication : 1 container chứa các article (table, view, stored procedure, UDF). Trong đó, phải chỉ rõ server :
 - + **Publisher** : chứa bản cơ sở dữ liệu gốc
 - Định nghĩa subscription (database): 1 container nhận publication. Trong đó, phải chỉ rõ server **Subscriber** : chứa cơ sở dữ liệu sau khi phân tán
3. Các hình thức phân mảnh : phân mảnh ngang (Horizontal), phân mảnh dọc (Vertical), phân mảnh hỗn hợp.

Trong phân mảnh ngang : nguyên thủy, dẫn xuất

Ví dụ: Giả sử ta muốn phân tán cơ sở dữ liệu QLDSV theo khoa thì nếu:

KHOA : MAKH='CNTT' → Khoa là phân mảnh ngang nguyên thủy

LOP : MAKH='CNTT' → LOP là phân mảnh ngang nguyên thủy

LOP.MAKH = KHOA.MAKH → LOP là phân mảnh ngang dẫn xuất

4. Các quy tắc phân mảnh: quá trình phân tán phải thỏa các qui tắc sau: **tính đầy đủ, tính tái thiết**, tính tách biệt

Ví dụ: Giả sử ta muốn phân tán cơ sở dữ liệu QLDSV theo 2 khoa (CNTT, VT) từ Server THU-PC đến 2 Server THU-PC\SERVER1 (chứa thông tin khoa CNTT) và Server THU-PC\SERVER2 (chứa thông tin khoa VT)

5. Tạo Link Server:

- Mục đích: cho phép truy cập dữ liệu từ server 1 đến Server 2 qua tập lệnh DML
- Cú pháp : Giả sử ta đang đứng ở Server THU-PC\SERVER1, ta muốn tạo 1 link server tên LINK1 đến Server THU-PC\SERVER2

```
EXEC sp_addlinkedserver @server='LINK1', @srvproduct='',
    @provider='SQLOLEDB', @datasrc=' THU-PC\ SERVER2'
```

Ví dụ áp dụng: Sử dụng link server vừa tạo để in ra danh sách lớp của khoa viễn thông đang đặt ở Server THU-PC\SERVER2

```
Select * from LINK2.QLDSV.DBO.LOP
```

6. Quy tắc để giúp cho các câu lệnh phân tán được “**trong suốt phân tán**”:

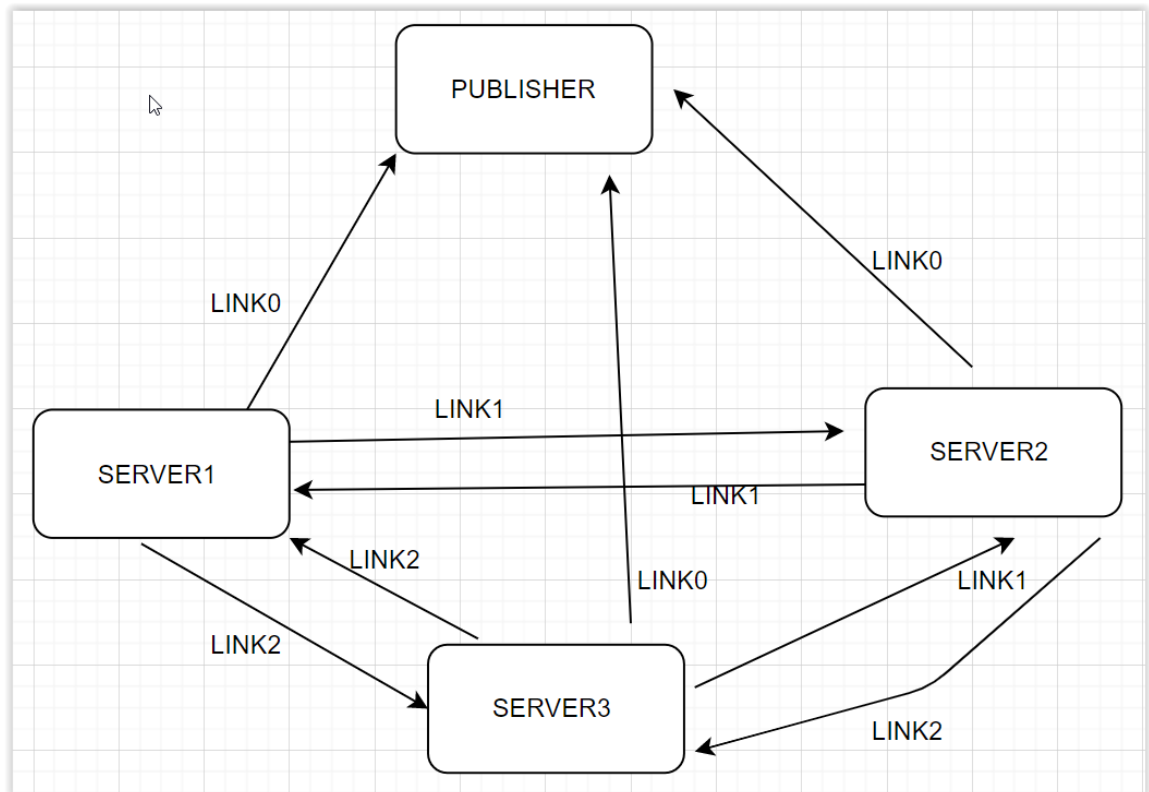
- Trong suốt : nghĩa là 1 SP khi ta cho thực thi ở 1 Server phân mảnh bất kỳ thì vẫn thực thi được ở tất cả các Server phân mảnh còn lại mà ta không cần chỉ lại đường dẫn đến SP cần truy xuất (người dùng không cảm nhận được là SP đang chạy trên Hệ thống phân tán).
- Quy tắc : điều kiện để hỗ trợ viết SP đạt được sự “trong suốt phân tán”
 - a. Tên các cơ sở dữ liệu ở các Server Subscriber phải giống nhau (trùng tên)
 - b. Tên Link Server phải giống nhau

- Cách tạo Link Server : LINK1 từ SERVER1 đến SERVER2

+ Ở SERVER2 : Tạo 1 login HTKN, pass, cho phép toàn quyền trên CSDL Subscription

+ Ở SERVER1 : tạo Link Server tên LINK1 nhờ login HTKN để kết nối đến SERVER2

Giả sử ta có 3 phân mảnh Server1, Server2, Server3 ngang cấp với nhau thì sẽ tạo các Link Server như sau:



7. Lệnh Select ... Union : lấy dữ liệu từ các server phân mảnh về
Ví dụ: Viết 1 SP tên DS_LOP sao cho khi chạy SP này trên 1 server bất kỳ , nó đều hiển thị danh sách lớp của trường

```
CREATE PROCEDURE SP_DSLOP
AS
SELECT * FROM
(SELECT MALOP, TENLOP, NK FROM DBO.LOP
UNION ALL
SELECT MALOP, TENLOP, NK FROM LINK1.QLDSV.DBO.LOP) TMP
ORDER BY NK
```

8. Tìm kiếm dữ liệu trên cơ sở dữ liệu phân tán:
Ví dụ: hãy in ra tên lớp và tên khoa, ho, ten của 1 sv có mã số là @X. (SP có tên sp_TimSV)

Lệnh kiểm tra SP vừa viết: Ta gọi lệnh sau ở cả 2 Server phân mảnh thì kết quả đều giống nhau, như vậy user sử dụng lệnh ta viết sẽ không cảm nhận được là cơ sở dữ liệu đang phân tán hay tập trung.

```
exec sp_TimSV 'ABC'
```

TÌM NV ở phân mảnh hiện tại trước, nếu không có thì về site chủ tìm.

```
create procedure sp_TimNV
    @X INT
AS
    DECLARE @MACN VARCHAR(10), @HO nvarchar(50), @TEN
    nvarchar(10)
    IF exists(select MANV from dbo.NHANVIEN where MANV =@X)
    BEGIN
        SELECT TENCN= (SELECT CHINHANH FROM dbo.CHINHANH), HO, TEN
        FROM NHANVIEN WHERE MANV=@X
    END
    ELSE
        IF exists(select MANV from LINK0.QL_VATTU.dbo.NHANVIEN
where MANV =@X)
        BEGIN
            SELECT @MACN= MACN, @HO=HO, @TEN=TEN
            FROM LINK0.QL_VATTU.dbo.NHANVIEN WHERE MANV=@X
            SELECT TENCN=CHINHANH, HO=@HO, TEN =@TEN
            FROM LINK0.QL_VATTU.dbo.CHINHANH WHERE MACN=@MACN

        END
    ELSE -- không có nv
        raiserror ( 'Ma nhan vien ban tim khong co', 16, 1)
```

9. Viết 1 SP cho phép sửa họ của 1 sv có mã sinh viên là @X, biết rằng ta chưa rõ sv này thuộc khoa nào ?

```
create procedure Update_Hosv
    @X char(10), @ho varchar(40)
AS
    if exists(select masv from sinhvien where masv =@X)
        Update sinhvien
        set Ho= @ho
        where masv=@X
    else
        if exists(select masv from link0.qlsv.dbo.sinhvien where masv
=@X)
        Update link0.qlsv.dbo.sinhvien
        set Ho= @ho
        where masv=@X
```

else

raiserror ('Ma sinh vien ban tim khong co', 16, 1)

Lệnh chạy để test SP :

exec update_hosv '96v10002', 'Nguyen van'

- **Viết 1 SP để xóa 1 sv có mã số @X**

create procedure Delete_Masv

@X char(8)

AS

if exists(select masv from sinhvien where masv =@X)

Delete from sinhvien

where masv=@X

else

if exists(select masv from link0.qlsv.dbo.sinhvien where masv
=@X)

Delete from link0.qlsv.dbo.sinhvien

where masv=@X

else

raiserror ('Ma sinh vien ban tim khong co', 16, 1)

GO

- **Viết 1 SP để chuyển 1 sv có mã số @X sang lớp có mã @malop**

create procedure ChuyenLop

@X nvarchar(8), @ML nvarchar(10)

AS

if exists(select masv from sinhvien where masv =@X)

if exists(select malop from Lop where malop =@ML)

Update sinhvien set malop=@ML where masv=@X

else

Print 'Ma lop' + @ML + 'khong co trong danh sach lop cua
khoa'

else

if exists(select masv from link0.qlsv.dbo.sinhvien where masv
=@X)

if exists(select malop from link0.qlsv.dbo.Lop where malop
=@ML)

Update link0.qlsv.dbo.sinhvien set malop=@ML where
masv=@X

else

Print 'Ma lop' + @ML + 'khong co trong danh sach lop cua
khoa'

else

print 'Error...'

- Viết SP_TimSV trả về tên lớp, tên khoa, họ tên của sinh viên dựa vào
mã sinh viên @X trong CSDL TRACNGHIEM

```
ALTER procedure sp_TimSV
    @X Nvarchar(10)
AS
    DECLARE @MAKH NVARCHAR(10), @MALOP NVARCHAR(10),
    @TENLOP NVARCHAR(50),
    @HO nvarchar(50), @TEN nvarchar(10)

    If exists(select masv from sinhvien where masv =@X)
    BEGIN
        SELECT @MALOP = MALOP, @HO=HO, @TEN=TEN
        FROM SINHVIEN WHERE MASV=@X
        SELECT @TENLOP = TENLOP , @MAKH = MAKH
        FROM LOP WHERE MALOP=@MALOP

        SELECT tenlop= @TENLOP,
            tenkh = (SELECT TENKH FROM KHOA WHERE MAKH=@MAKH),
            HO=@HO, TEN=@TEN
    END
    ELSE
    If exists(select masv from LINK0.TRACNGHIEM.dbo.sinhvien where
    masv =@X)
    BEGIN
        SELECT @MALOP = MALOP, @HO=HO, @TEN=TEN
        FROM LINK0.TRACNGHIEM.dbo.SINHVIEN WHERE MASV=@X
        SELECT @TENLOP = TENLOP , @MAKH = MAKH
        FROM LINK0.TRACNGHIEM.dbo.LOP WHERE MALOP=@MALOP

        SELECT tenlop= @TENLOP,
            tenkh = (SELECT TENKH FROM LINK0.TRACNGHIEM.dbo.KHOA
            WHERE MAKH=@MAKH),
            HO=@HO, TEN=@TEN
    END
    ELSE RAISERROR (N'Mã SV không có trong DS',16,1)
```

```

ALTER PROCEDURE TIMPHIEU
@X CHAR (10)
AS
    if exists(select PHIEU from link1.QLVT.dbo.PHATSINH where
PHIEU =@X)
        select PS.PHIEU,HOTENKH,NGAY,LOAI,
TENVT,SOLUONG,DONGIA,TRIGIA=SOLUONG*DONGIA
        from link1.QLVT.dbo.PHATSINH PS ,
link1.QLVT.dbo.CT_PHATSINH CTPS ,
        link1.QLVT.dbo.VATTU VT
        where PS.PHIEU=CTPS.PHIEU AND CTPS.MAVT=VT.MAVT
        and PS.PHIEU=@X
    ELSE
    if exists(select PHIEU from link2.QLVT.dbo.PHATSINH where
PHIEU =@X)
        select PS.PHIEU,HOTENKH,NGAY,LOAI,
TENVT,SOLUONG,DONGIA,TRIGIA=SOLUONG*DONGIA
        from link2.QLVT.dbo.PHATSINH PS ,
link2.QLVT.dbo.CT_PHATSINH CTPS ,
        link2.QLVT.dbo.VATTU VT
        where PS.PHIEU=CTPS.PHIEU AND CTPS.MAVT=VT.MAVT
        and PS.PHIEU=@X
    ELSE
    print 'Ma PHIEU ban tim khong co'

```

```

CREATE FUNCTION FN_TIMNV_ (@MANV nvarchar(10) )
RETURNS TABLE
AS

RETURN
(
    SELECT HOTEN=HO+ ' ' +TEN , TENCN = (SELECT CHINHANH FROM dbo.CHINHANH)
    FROM NHANVIEN WHERE MANV = @MANV
)

ALTER PROC [dbo].[SP_TIMNV]
@MANV nvarchar(10)
AS
BEGIN

    declare @q CHAR(1)= ' '
    declare @s nvarchar(4000)

    SET @s= N'SELECT HOTEN , TENCN FROM FN_TIMNV_ (' + @MANV + ') UNION ' +
        'SELECT HOTEN , TENCN FROM OPENQUERY(LINK1,' + @q +
            'SELECT HOTEN , TENCN FROM QLVT_D14CP.DBO.FN_TIMNV_
(' +
            @manv+ ' ')+@q + ' )'
    --select @s
    EXEC sp_executesql @s

END

ALTER PROC [dbo].[SP_TIMNV]
@MANV INT
AS
IF EXISTS (SELECT MANV FROM NHANVIEN WHERE MANV=@MANV )
    SELECT TENCN =(SELECT CHINHANH FROM CHINHANH ),
        HOTEN =HO+ ' ' +TEN
    FROM NHANVIEN WHERE MANV=@MANV
ELSE
IF EXISTS (SELECT MANV FROM LINK1.QLVT_D14IS.DBO.NHANVIEN WHERE MANV=@MANV
)
    SELECT TENCN =(SELECT CHINHANH FROM LINK1.QLVT_D14IS.DBO.CHINHANH ),
        HOTEN =HO+ ' ' +TEN
    FROM LINK1.QLVT_D14IS.DBO.NHANVIEN WHERE MANV=@MANV
ELSE
    RAISERROR ('Mã nhân viên không có .',16,1)

CREATE PROC SP_Lay_Thong_Tin_NV_Tu_Login
@TENLOGIN NVARCHAR( 100)
AS
DECLARE @UID INT
DECLARE @MANV NVARCHAR(100)
SELECT @UID= uid , @MANV= NAME FROM sys.sysusers
    WHERE sid = SUSER_SID(@TENLOGIN)

SELECT MAGV= @MANV,
    HOTEN = (SELECT HO+ ' ' +TEN FROM dbo.NHANVIEN WHERE MANV=@MANV ) ,

```

```
TENNHOM=NAME  
FROM sys.sysusers  
WHERE UID = (SELECT groupuid FROM sys.sysmembers WHERE memberuid=@uid)
```


View Get_Subscribes cài đặt trên CSDL gốc (Publisher) để lấy về danh sách các phân mảnh, và các server Subscriber tương ứng

```
CREATE VIEW Get_Subscribes
```

```
AS
```

```
SELECT TENCN=PUBS.description, TENSERVER=subscriber_server  
FROM sysmergepublications PUBS, sysmergesubscriptions SUBS  
WHERE PUBS.pubid = SUBS.pubid AND publisher <>  
subscriber_server
```

```
GO
```