1. **Dữ liệu rất lớn và đặt ở nhiều nơi thì tại sao chúng ta lại nghiên cứu truy vấn trên một môi trường db?**

Bởi vì khi xử lý trên hệ thống phân tán ta cũng phải xử lý trên từng server. Nếu trên từng db ta đã tiết kiệm thời gian và tối ưu hóa thì tổng thời gian tối ưu như vậy ta có tổng thời gian cuối cùng là những min nhỏ nhất của từng db thì sẽ ra đc thời gian nhỏ nhất của từng hệ thống khi xử lý bài toán đó.  
Nhưng còn cách khác là thay vì đi qua tuần tự từng site đó thì có phương án khác là xử lý song song. Tại 1 thời điểm nào đó đưa ra các lệnh gửi tới các site để chạy cùng lúc.

1. **1 csdlpt là tập hợp dữ liệu của 1 hệ thống nhưng được phân bố trên nhiều địa chỉ (site) của 1 mạng máy tính. Có 2 khía cạnh quan trọng:**

* Sự phân tán: dữ liệu ko lưu trữ trên cùng 1 địa điểm vì thể có thể phân biệt nó với csdl tập trung
* Mối tương quan luận lý: Các dữ liệu có 1 số thuộc tính ràng buộc với nhau từ các csdl cục bộ mà đc lưu trữ tại các địa điểm khác nhau trên mạng

1. **Ứng dụng phân tán là gì?**

Là các ứng dụng mà truy xuất dữ liệu đang đặt ở nhiều nhánh (ít nhất là 2)

***Lưu ý:*** những câu chưa có => là những câu chưa có đáp án chính xác!

1. **Khi nào chúng ta cần phải tạo link server? Tạo link server cần các ràng buộc là gì?** => tạo link server khi database nằm ở 2 server khác nhau, và ràng buộc là: tên các server link tới nhau thì phải cùng tên. Khi link server, link càng nhỏ thì càng gần
2. **Thế nào là tính trong suốt của 1 SP? Điều kiện để biết 1 SP trong suốt? Điều kiện để viết 1 SP trong môi trường phân tán? Thế nào là SP trong suốt?** => Điều kiện để viết SP tên database ở các phân mảnh phải giống nhau, tên link server các server phải giống nhau. SP trong suốt là SP khi chạy trên tất cả các phân mảnh phải cho ra các kết quả như nhau. (tài liệu không chỉ ra tính trong suốt của 1 SP và điều kiện để biết 1 SP trong suốt, chỉ có 2 câu sau là biết câu trả lời)
3. Giao tác là gì? Giao tác phân tán là gì? Cho biết ứng dụng và ý nghĩa của giao tác phân tán? (chưa có câu trả lời)
4. **Phân mảnh ngang là gì? Nhìn vào database của mình chỉ ra đâu là phân mảnh ngang nguyên thủy, đâu là phân mảnh ngang dẫn xuất?** => Phân mảnh ngang là phân mảnh chia một quan hệ theo các bộ. Ví dụ: đề tài trắc nghiệm, phân tán theo cơ sở thì ngang nguyên thủy là ở table Cơ sở, điều kiện để ràng buộc phân mảnh ngang nguyên thủy đó là: Coso.Macs = khoa.Macs. Tương tự cho các đề tài vật tư hay sinh viên. Vật tư thì: CHINHANH.MACN = NHANVIEN.MACN và bảng chi nhánh là ngang nguyên thủy. (tài liệu không chỉ ra phân mảnh dẫn xuất)
5. **Quy trình thực hiện tối ưu hóa truy vấn trong cơ sở dữ liệu phân tán?** => Thực hiện: chọn trước, chiếu trước, kết sau. Lượt bỏ và tối ưu trong lệnh WHERE bằng việc thực hiện các phép biến đổi tương đương.
6. **Nhìn database cho biết đề tài mình làm đâu là nhân bản? Nhân bản để làm gì**? => Đề trắc nghiệm thì nhân bảng có 3 bảng: bộ đề, giáo viên, môn học,… vật tư thì có: kho... Những bảng được nhân bản là những bảng được sử dụng nhiều trong mục đích select, ít thêm, xóa, sửa! Vì vậy ta nhân bản để khi truy xuất dữ liệu nhiều sẽ có luôn trên các site đó mà không cần phải link qua site khác để tìm kiếm, tốn xử lý và time!
7. **Tại sao khi ta gõ login vào đăng nhập là biết được họ tên và thông tin người đăng nhập** => lấy theo mã trong database role sẽ truy xuất đến bảng mã nhân viên (đối với vật tư) hay mã giáo viên (đối với đề tài trắc nghiệm)
8. **Để viết giao tác, ta cần phải bật dịch vụ nào?** => Vào windows + R -> services.msv -> start Distributed Transaction Coordinator
9. **Câu lệnh bắt đầu và kết thúc của một giao tác là gì? =>** begin distributed Transaction >> bắt đầu … kết thúc >> commit transaction
10. **Tính chất của giao tác:** 4 tính chất: nguyên tử (thành công nếu tất cả thành công), nhất quán (vẫn đảm bảo các ràng buộc), tỉnh cô lập (Isolation), tính bền vững (thể hiện mức độ riêng tư của từng giao tác, khi hệ thống có lỗi nhưng thành công của giao tác vẫn được ghi vào data)
11. **Database sau khi phân tán phải đảm bảo những tính chất gì?** => đảm bảo 3 tính chất sau: tính đầy đủ, tính duy nhất - tách biệt, tính tái thiết được. (thầy sẽ hỏi thêm 1 câu nữa: vậy đề tài của em chỉ ra bảng nào vi phạm 1 trong 3 tính trên >>> thì ngay lập tức trả lời là những bảng nào là nhân bản thì sẽ vi phạm tính duy nhất (tách biệt)
12. **Có bao nhiêu mức trong suốt? Và mức nào người lập trình hay sử dụng**? => có 4 mức trong suốt: Trong suốt phân tán (dữ liệu được phân tán xuống và tên các database phân tán giống nhau), trong suốt định vị (khi gọi SP đến các phân mảnh này chỉ cần gọi tên mà không cần gọi ra vị trí của bảng đó), trong suốt ánh xạ cục bộ (viết SP link đến phân mảnh phải gọi ra tên và vị trí cấp phát của nó), không trong suốt. Người lập trình sẽ chọn 3 mức để làm trong suốt nhưng sẽ làm cho người người sử dụng có cảm giác đang sử dụng cơ sở phân tán như mức 1.
13. **Quy trình thực hiện phân tán?** Tạo folder share (read và write) >> tạo publication >> add article >> Nói chung là những bước phân tán làm những gì thì kể ra hết, tài liệu ghi rất rõ >> tạo subcription >> chọn server cần phân tán.
14. Các hàm lấy dữ liệu, show dữ liệu, các khái niệm như subform là gì cũng được thầy hỏi.
15. **Các bước phân tán? Khi phân tán cần bật dịch vụ gì?** Cách bật như thế nào?

* Các bước phân tán:
  + Cấu hình các server:
    - Publisher: chứ bản cơ sở dữ liệu gốc
    - Distributor: điều phối các mảnh phân tán
    - Subscriber: chứa cơ sở dữ liệu sau khi phân tán
  + Định nghĩa publication: 1 container chứa actical (view, SP, table, UDF).
  + Định nghĩa subcription (database) 1 container nhận publication.
* Khi phân tán cần bật dịch vụ SQL Server Agent. Cách bật Program -> Sql2008 -> configration tool -> bật dịch vụ Agent.

1. **Thuộc tính của folder RELPDATA là gì? Ở đâu? Mục đích của folder này dùng để làm gì?** => thuộc tính của folder RELPDATA là share, có quyền read/write. Mục đích của folder này để chứa các dữ liệu trao đổi trong quá trình update dữ liệu trao đổi trong quá trình update dữ liệu từ các phân mảnh về cơ sở dữ liệu gốc, và từ cơ sở dữ liệu gốc đến các phân mảnh.
2. **Các khai niệm publication, subscription, publisher, subscriber, distributor?** => Publisher (server gốc): chứa dữ liệu gốc; Subscription (server phân mảnh): chứa dữ liệu sau khi phân mảnh; Distribution: điều phối các mảnh phân tán
3. **Actical bao gồm những gì?** => actical bao gồm view, table, SP, UDF
4. **Phân mảnh ngang là gì? Phân mảnh dọc là gì? Phân mảnh ngang nguyên thủy là gì? Phân mảnh ngang dẫn xuất là gì**? =>

* Phân mảnh ngang (phép hội union) là chia 1 qan hệ theo các bộ. Mỗi phân mảnh ngang là tập con của quan hệ. Có hai loại phân mảnh ngang: phân mảnh ngang nguyên thủy và phân mảnh ngang dẫn xuất.
  + Trong đó phân mảnh ngang nguyên thủy: 1 quan hệ dựa trên 1 vị từ (là 1 điều kiện từ trên 1 giá trị hằng) định nghĩa trên 1 quan hệ.
  + Phân mảnh ngang dẫn xuất: một quan hệ dựa trên 1 vị từ định nghĩa trên 1 quan hệ khác. Vd: chương trình quản lý điểm sinh viên phân mảnh ngang theo khoa nếu: KHOA.MAKH = “CNTT” -> khoa là phân mảnh ngang nguyên thủy. LOP.MAKH = “CNTT” -> lớp là phân mảnh ngang nguyên thủy. KHOA.MAKH = LOP.MAKH -> lớp là phân mảnh ngang dẫn xuất.
* Phân mảnh dọc (phép kết tự nhiên có khóa chính): chia các thuộc tính vào hai nhóm, các mảnh nhận được từ phép chiếu quan hệ trên mỗi nhóm. Phân mảnh hỗn hợp là sự kết hợp của hai phân mảnh trên (rất ít hỏi)

1. Nhân bản là gì? Lý do nhân bản? Nhân bản không hiệu quả với lệnh gì? (không có câu trả lời)

- Lý do: select dữ liệu nhanh hơn khi nó được nhân bản từ server khác

- Nhân bản không hiệu quả với các lệnh insert, delete, update

1. Trong lược đồ phân mảnh, số lượng database trong publication và số lượng database trong subscriber như thế nào? => số lượng database trong publication nhiều hơn trong subscrition vì trong publication chứa nhiều subscrition
2. Lý do nào người ta định nghĩa 1 publication và 1 subscriber?
3. Trong database cái nào là nhân bản, cái nào là phân hoạch? (xem điều kiện) => môn học là nhân bản, còn lại là phân hoạch.
4. Sau khi định nghĩa publication và subscriber ta phân tán thì server chứa publication gọi là publisher server subscrition chứa subcriber server, còn lại là distibution.
5. Nếu muốn phân tán database từ máy này sang máy kia thì subscriber chứa gì? (không có câu trả lời)
6. **Nếu phân tán xong csdl, muốn thay đổi cấu trúc (cột) của table server gốc thì ta làm thế nào? =>** có thể thay đổi cột bằng cách viết SP dùng lệnh ALTER TABLE ở server gốc sau khi thêm xong thì đẩy xuống phân mảnh con.
7. **Lệnh lấy dữ liệu từ lớp hiện về sub form như thế nào? Sự kiện gì?** (đối với những người dùng dataset)
8. **Đưa ra 2 cách xử lý truy vấn dữ liệu cùng lúc ở 2 server khác nhau** => dùng câu truy vấn song song cho nó chạy song song ở hai server.
9. **Vì sao phân mảnh dọc lại cần có khóa chính?** => Để tái thiết được dữ liệu.

INTERVIEW QUESTIONS:

1. **Sau khi phân tán xong thì có 1 trường dữ liệu là rowguild. Vậy rowguild được sinh ra để làm gì?**

Hỗ trợ quá trình đồng bộ dữ liệu từ site phân mảnh về site chủ và ngược lại

1. **Login Name là gì? Username là gì?**

Login Name là tài khoản chúng ta dùng để đăng nhập vào 1 server. Ví dụ như tài khoản SA – tài khoản phổ biển khi đăng nhập SQL Server để tăng độ bảo mật. Sau khi kết nối thành công tới 1 server, chọn Security => Logins sẽ thấy tài khoản SA

Username là tài khoản mà chúng ta dùng để làm việc trên 1 csdl. Ví dụ trên cơ sở dữ liệu QLVT, chọn vào QLVT => Security => Users sẽ thấy các tài khoản có thể làm việc trên csdl này

1. **Db\_dataReader, db\_dataWriter, db\_securityAdmin,… rồi db\_accessAdmin là gì? Chúng để làm gì?**

Chúng là những quyền mà một tài khoản kiểu login name có thể đc chỉ định và quyết định xem chúng có thể làm những gì trên csdl đó

+ db\_dataReader là quyền chỉ xem dữ liệu, ko thể thêm mới hoặc sửa đổi dữ liệu

+ db\_dataWriter là quyền chỉ ghi dữ liệu, di nhiên nếu ghi được thì coi như cũng đọc được dữ liệu

+ db\_securityAdmin là quyền tạo các tài khoản login name, tài khoản này dùng để đăng nhập vào các server

+ db\_accessAdmin là quyền tạo các tài khoản username, tài khoản dùng để thao tác trên 1 csdl nhất định

+ db\_owner là quyền cao nhất với 1 csdl, quyền này cho phép xem, thêm, xóa, sửa tất cả dữ liệu và tạo tài khoản login name và username mà ko có giới hạn nào

1. **Có 2 cách để viết SP. Cách 1 đã đc trình bày ở mục SP. Cách viết số 2 là viết SP thủ công tại từng phân mảnh.** Cách làm nào hiệu quả hơn?

Cách làm thứ nhất hay hơn. Nếu ta chỉnh sửa SP trên server gốc thì sự thay đổi sẽ đc tự động cập nhật xuống các server phân mảnh. Còn cách thứ 2 thì phải sửa thủ công trên từng phân mảnh.

1. **Nếu sửa SP trên phân mảnh thì nó có đồng bộ về các phân mảnh khác và server gốc, điều này là đúng hay sai?**

Sai. Code chỉ đồng bộ theo 1 chiều từ server gốc tới các phân mảnh. Không có chiều ngược lại

1. **Sửa dữ liệu tại server gốc thì phân mảnh có nhận được không? Nếu ngược lại, từ server phân mảnh về server gốc thì có nhận được không?**

Có. Dữ liệu đồng bộ theo 2 chiều

1. **Nêu ưu và nhược điểm khi ưu tiên tìm kiếm trên site phân mảnh trước khi về site chủ**

Ưu điểm: Server gốc lúc nào cũng hoạt động nên luôn truy vấn được dữ liệu mong muốn   
Nhược điểm: Không thể che dấu được hoàn toàn các thông tin nhạy cảm giữa các site phân mảnh. Việc cấp 1 tài khoản để vào site chủ có thể giúp người dùng đó xem được dữ liệu của các site khác

1. **Muốn thực thi 1 câu lệnh SP, View, Function thì ta làm ntn?**

Ta sẽ tạo 1 đối tượng là SqlCommand rồi nhúng các đối số cần thiết vào

1. **Giao tác là gì ? Để viết giao tác cần bật dịch vụ gì ?**

Giao tác là 1 dãy các thao tác đọc và ghi dữ liệu cùng với các bước tính toán nhất quán (Begin Trans, Commit, Rollback, Begin distributed trans) để giải quyết các tình huống khi dữ liệu bị mất tính nhất quán khi có nhiều truy xuất đồng thời   
Để viết giao tác cần phải bật dịch vụ MSDTC – Microsoft Distribute Transaction Coordinator

1. **Ý nghĩa của Begin trans, Commit, Rollback và Begin distributed trans là gì?**

+ Begin Trans: bắt đầu giao tác

+ Commit: Xác nhận thành công

+ Rollback (hoặc Abort): Hủy bỏ giao tác và trả lại dữ liệu cũ

+ Begin distributed trans: Câu lệnh mở đầu giao tác phân tán

1. **Nêu những tính chất giao tác ?**

Có tất cả 4 tính chất giao tác bao gồm:

+ Tính nguyên tử: 1 giao tác có nhiều tập lệnh thì mọi câu lệnh hoặc thành công hoặc nếu 1 lệnh thất bại thì tất cả thất bại

+ Tính nhất quán: đảm bảo tất cả ràng buộc. Mọi lệnh đều đc chạy

+ Tính biệt lập: 1 giao tác đang thực thi thì ko làm lộ các kết quả của nó cho những giao thức khác đang cùng hoạt động trước khi nó hoàn thành

+ Tính bền vững: 1 giao tác đã đc commit thì dữ liệu đó đc giữ nguyên và ko thể rollback

1. **Dữ liệu rác là gì?**

Là dữ liệu sinh ra có thể chưa phải là dữ liệu đã đc commit nếu 1 giao tác khác đồng thời vào và lấy giao tác này ra thì dữ liệu này là ko chính xác

1. **Có mấy loại giao tác ?**

Có 2 loại giao tác đó là giao tác tập trung và giao tác phân tán:

Giao tác tập trung có 2 loại: giao tác phẳng và giao tác lồng

+ Giao tác phẳng có điểm khởi đầu duy nhất (begin transaction) và 1 điểm kết thúc duy nhất (end transaction)

+ Giao tác lồng chứa nhiều giao tác với điểm khởi đầu và duy nhất riêng biệt

Giao tác phân tác chỉ có 1 loại duy nhất: giao tác phẳng

1. **XACT\_ABORT là gì ? Nó nhận đc mấy giá trị ?**

XACT\_ABORT: đây là tùy chọn kết nối. Nó chỉ nhận 2 giá trị là on và off.

Nếu là Off, SQL Server sẽ bỏ qua lệnh gây lỗi transaction và tiếp tục thực thi các lệnh còn lại.

Nếu là On, SQL Server sẽ hủy bỏ toàn bộ transaction nếu nó gặp lệnh bị lỗi và trả lại dữ liệu về ban đầu.

1. **Khi viết 1 SP khi nào ta không dùng begin transaction, không dùng commit,.. vẫn đc coi là 1 giao tác ?**

Khi ta sử dụng chỉ 1 lệnh UPDATE – DELETE – INSERT duy nhất. Còn nếu có 2 lệnh UPDATE – DELETE – INSERT thì phải dùng cú pháp begin transaction và commit.

1. **Dịch vụ MS TDC là gì ?**

Dịch vụ MS TDC là Microsoft Distributed Transaction Coordinator. Nó đảm bảo tất cả cập nhật đc thực thi ở tất cả các phân mảnh hoặc trong trường hợp lỗi thì xem như chưa thực hiện thao tác nào hết.

1. **Vị từ thích hợp là gì ?**

Thỏa tính đầy đủ và tính cực tiểu

1. **Tiêu chí đẩy đủ và tiêu chí cực tiểu là gì ?**

Tiêu chí đầy đủ là ta có nhiều SP thì xác suất nó truy xuất tới các phân mảnh là như nhau

Tiêu chí cực tiểu là mỗi site đc sinh ra phải đc sử dụng trong 1 SP nào đó

1. **Vị từ là gì ? 1 vị từ đơn giản là gì ?**

Vị từ là mệnh đề logic có nhiều điều kiện

1 vị từ đơn giản là vị từ có kiểu: MACN = ‘CN1’, … Tức vế trái là tên thuộc tính và vế phải là tên giá trị

Một vị từ sơ cấp: MACN = (CN1 hội CN2)

1. **1 vị từ “thích hợp” là gì ?**

1 vị từ “thích hợp” nghĩa là khi tạo 1 phân mảnh “thích hợp” thì phải có 1 SP sử dụng tới phân mảnh này

**21. Sự trong suốt phân tán là gì?**

Đáp án: Sự trong suốt phân tán: khi 1 SP được thực thi ở 1 server phân mảnh bất kỳ thì cũng hoạt động tốt trên các phân mảnh còn lại - người dùng không cảm thấy được sp đang chạy trên hệ thống phân tán.

Chú ý: chúng ta đứng ở vị trí thứ 3 để lập trình, luôn ưu tiên tìm ở phân mảnh cục bộ trước khi sang các phân mảnh khác

**22. Có mấy mức độ trong suốt phân tán là gì?**

Đáp án: có 4 mức độ trong suốt phân tán

1. Mức cao nhất: không cần chỉ ra phân mảnh cần truy vấn. Người dùng không cảm giác là đang thao tác trên một câu truy vấn phân tán.
2. Sự trong suốt vị trí: chúng ta chỉ cần chỉ ra tên TABLE cần phải truy vấn.
3. Sự trong suốt ánh xạ cục bộ: Chúng ta cần chỉ rõ tên phân mảnh và vị trí cấp phát của chúng
4. Không trong suốt: chúng ta phải viết lệnh để đến đúng vị trí database cần thao tác. Tuy nhiên, do chúng ta đã đặt tên các cơ sở dữ liệu giống nhau ở mọi phân mảnh. Tên LINK server cũng trùng tên nên không phải bận tâm đến mức này.

Lưu ý: Người dùng sẽ sử dụng ứng dụng ở mức cao nhất. Còn người lập trình như chúng ta sẽ đứng ở mức thứ 3

**23a. Điều kiện để có thể phân tán được cơ sở dữ liệu? Có mấy bước để phân tán cơ sở dữ liệu**? Đáp án: có dịch vụ SQL Server Agent

Có 3 bước để thực hiện phân tán cơ sở dữ liệu

1. Định nghĩa Server Distributor: chứa Database distribution
2. Định nghĩa publication: 1 container chứa các article (table, view, SP, UDF).

Trong đó, phải chỉ rõ các server:

Publisher – chứa bản gốc cơ sở dữ liệu

Distributor - điều phối phân tán dữ liệu

1. Định nghĩa subscription (database): 1 container nhận publication. Trong đó, phải chỉ rõ server Subscriber - chứa cơ sở dữ liệu sau khi phân tán

**23b. Có mấy hình thức phân mảnh?**

Đáp án: có 3 hình thức phân mảnh, gồm: phân mảnh ngang, phân mảnh dọc và phân mảnh hỗn hợp

**24a. Nêu đặc điểm của phân mảnh ngang?**

Đáp án: Phân mảnh ngang là chia một quan hệ theo các bộ . Mỗi phân mảnh ngang là tập hợp con của quan hệ.

Trong phân mảnh ngang bao gồm phân mảnh ngang nguyên thủy và phân mảnh ngang dẫn xuất.

Phân mảnh ngang nguyên thủy là phân mảnh của 1 quan hệ dựa trên 1 vị từ được định nghĩa trên quan hệ đó

Phân mảnh ngang dẫn xuất là phân mảnh của 1 quan hệ dựa trên vị từ được định nghĩa trên quan hệ khác

Ví dụ, với đề quản lý vật tư thì

MaChiNhanh = ‘CN1’ -> phân mảnh ngang nguyên thủy

NHANVIEN.MaKho = KHOHANG.MaKho -> phân mảnh ngang dẫn xuất

Nói cách khác ‘CN1’ và ‘CN2’ là các vị từ.

**24b. Nêu đặc điểm của phân mảnh dọc ?**

Đáp: phân mảnh dọc là phân mảnh dựa trên khóa chính của một quan hệ (phải có khóa chính để đảm bảo tính tái thiết)

giả sử, NHANVIEN(id, ho, ten, diachi, luong, ngaysinh, ma chi nhanh) thì khi tạo phân mảnh dọc sẽ thành

NHANVIEN(id, ho, ten, ngay sinh, ma chi nhanh) trong đó MaChiNhanh là khóa ngoại thì phải cho phép NULL.

**25a. Nêu đặc điểm của phân mảnh hỗn hợp?**

Đáp: là sự kết hợp của 2 cách phân mảnh ngang và dọc. Khi hợp các phân mảnh theo 2 cách này thì cơ sở dữ liệu tạo thành chính là cơ sở dữ liệu ban đầu

**25b. Có mấy quy tắc phân mảnh?**

Đáp: có 3 quy tắc phân mảnh:

Tính đầy đủ: xét trên góc độ là QUAN HỆ. Mỗi một dữ liệu ít nhất phải có một phân mảnh thì không bị mất thông tin

Tính tái thiết: thường thỏa mãn tính đầy đủ sẽ thỏa mãn tính tái thiết.

Tính tách biệt: mỗi một dữ liệu chỉ nằm ở một phân mảnh duy nhất. Khi ghép các phân mảnh thì database sẽ đầy đủ như ban đầu. Tức nếu TEN ở phân mảnh 1 thì sẽ không xuất hiện ở phân mảnh 2

Lưu ý Thực tế, tính tách biệt dễ bị vi phạm vì NHÂN BẢN giúp dữ liệu tồn tại ở cả 2 phân mảnh. Bắt buộc tính đầy đủ phải thỏa mãn nếu thỏa mãn thì tính tái thiết cũng sẽ được bảo đảm.

**26: Run on continue khác run on demand ở điểm nào?**

Đáp: Run on continue làm tính nhất quán cao, dữ liệu đồng bộ ngay lập tức. Run on demand tính từ quản cao. Các site có thể disconnect. Các thay đổi không phản ảnh tức thời tới site chủ.

**27: snapshot folder là gì?**

Đáp: Folder chứa dữ liệu trung gian để đồng bộ dữ liệu từ các phân mảnh về site chủ và ngược lại. là folder chứa dữ liệu để qua đẩy lại. Phải là network path (shared folder).

**28: Tại sao trả về mã nhân viên khi dùng sp\_DangNhap?**

Đáp: để gán tự động cho các form có mã nhân viên

**29: 4 thuộc tính quan trọng của combo box?**

Data source chứa danh sách dữ liệu cung cấp cho nó

Display member chứa tên field mình muốn hiển thị

Value member chứa field dữ liệu mình muốn lấy

Selected index change phương thức thay đổi giá trị

**30. Vì sao phải dùng Remote Login?**

Đáp: cho phép truy cập dữ liệu khi đứng từ server này sang server khác.

**31: Có những cách nào để tối ưu hóa truy vấn?**

Đáp: Có 5 cách để tối ưu hóa một câu truy vấn

1. Dùng phép chọn, chiếu trước, phép kết sau.

2. Khử phép kết (nếu được)

3. Nếu 1 điều kiện xuất hiện nhiều lần trong WHERE thì dùng các phép biến đổi tương đương để cho điều kiện đó xuất hiện 1 lần

4. Trong mệnh đề AND, điều kiện nào có xác suất sai cao thì đặt ở đầu; OR thì ngược lại.

5. Field tham gia trong điều kiện truy vấn nên được sắp theo thứ tự trước và thứ tự này phải được sử dụng trong mệnh đề truy vấn với WITH (INDEX = ten\_index)

**32. Thế nào là một SP trong suốt? Điều kiện để viết SP trong suốt?**

Đáp: SP được coi là trong suốt nếu ta cho thực thi ở 1 server thì vẫn cho thực thi được ở những server còn lại mà không cần chỉ rõ đường dẫn đến table cần truy xuất

Hoặc có thể trả lời

SP trong suốt là SP mà khi ta cho thực thi ở bất kỳ phân mảnh nào đều cho kết quả giống nhau

Điều kiện để viết SP trong suốt là ta cần có LINK SERVER & tên database giống nhau

**33. Trong database, cái nào là nhân bản, cái nào là phân hoạch?**

Đáp: mở cây dẫn xuất ra xem, cái nào không có trong cây dẫn xuất là nhân bản, cái đầu tiên là phân mảnh nguyên thủy.

**34. Nếu đã phân tán xong cơ sở dữ liệu, muốn thay đổi cấu trúc hoặc thứ tự các cột trong table của server gốc thì làm sao?**

Đáp: có thể thay đổi bằng cách viết SP dùng lệnh ALTER TABLE ở server gốc sau đó đồng bộ xuống các server phân mảnh

**35. Dữ liệu sau khi nhập form sẽ được đẩy về đâu?**

Đáp: đẩy về publisher sau đó đồng bộ xuống các subcirber

**36. Trong các table, cái nào mang tính đầy đủ, cái nào vi phạm tính tách biệt?**

Đáp: các table nhân bản thì vi phạm tính tách biệt, tất cả các table còn lại thì mang tính đầy đủ.

**37. Giao tác tập trung với giao tác phân tán giống và khác nhau như thế nào?**

Đáp:

Điểm khác: giao tác tập trung thì thực thi trên môi trường cơ sở dữ liệu tập trung (gồm có giao tác phẳng và giao tác lồng), còn giao tác phân tán thì thực thi trên môi trường cơ sở dữ liệu phân tán.

Điểm giống: 4 tính chất của giao tác

**38. Login name nằm trong table nào?**

Đáp: nằm trong table sys.sysuser trong database đó.

**39. Tại sao biết user liên kết với login nào?**

Đáp: username và login name liên kết với nhau qua trường sid (trên user và login đều có trường sid) nên từ login name biết được username từ sid.

**40. Tên nhóm quyền nằm ở table nào?**

Đáp: nằm trong table sys.sysuser trong database đó.

**41. Ưu khuyết điểm của nhân bản?**

Đáp: Truy xuất nhanh, đứng ở đâu cũng có thể select được. Nhưng update chậm vì có quá nhiều bản sao.

**42a. Ưu khuyết điểm của phân hoạch?**

Đáp: select chậm nhưng insert và update nhanh do chỉ thao tác trên server gốc hoặc server phân mảnh cần truy xuất.

**42b. Tại sao table đó nhân bản?**

Đáp: Trong đề vật tư thì vật tư nhân bản vì vật tư có thể ở chi nhánh này cũng có thể ở chi nhánh khác. Trong đề trắc nghiệm thì giáo viên nhân bản vì giáo viên có thể dạy ở cơ sở này, có thể dạy ở cơ sở khác, bộ đề đi theo giáo viên nên cũng nhân bản. Giáo viên ở cơ sở 1 có thể dạy cho cơ sở 2 nhưng không thể đăng ký thi cho lớp ở cơ sở 2.

**PHẦN I: PHÂN TÁN**

**Câu 1: Các bước phân tán? Khi phân tán cần bật dịch vụ gì? Cách bật như thế nào?**

* Các bước phân tán:

+ Cấu hình server:

* Publisher: chứa bản góc của csdl
* Distributor: điều phối các mãnh phân tán
* Subscriber: chứa csdl sau khi phân tán

+ Định nghĩa Publication: 1 container chứa Actical(view, sp, table, UDF).

+ Định nghĩa Subsciption: 1 container nhận publication.

* Khi phân tán phải bật dịch vụ SQL Server Agent
* Cách bật: run gõ services.msc-> start SQL Server Agent

**Câu 2: thuộc tính folder RELPDATA là gì? ở đâu? Mục đích của thư mục để làm gì?**

* Thuộc tính là share, có quyền read/write. Mục đích để chứa các dữ liệu trao đổi trong quá trình update dữ liệu từ các phân mảnh về csdl gốc , và từ gốc đi đến các phân mảnh

**PHẦN II: SỰ TRONG SUỐT PHÂN TÁN**

**Câu 1: Phân mãnh ngang là gì? Phân mãnh dọc là gì? Phân mãnh ngang nguyên thủy là gì? Phân mãnh ngang dẫn xuất là gì?**

* Phân mảnh ngang: là chia 1 quan hệ theo các bộ. Mỗi phân mảnh ngang là một tập con của quan hệ. Có hai loại phân mảnh ngang đó là phân mảnh ngang dẫn xuất và phân mảnh ngang nguyên thủy.
  + Phân mảnh ngang nguyên thủy là sự phân mảnh dựa trên một qua hệ vị từ được định nghĩa trên một quan hệ
  + Phân mảng ngang nguyên thủy là sự phân rả một quan hệ dựa vào các vị từ được định nghĩa trên một quan hệ khác.
* Sự phân mảnh dọc của một quan hệ toàn cục là việc chia các thuộc tính vào hai nhóm; các mảnh nhận được từ phép chiếu quan hệ toàn cục trên mỗi nhóm

**Câu 2: Nhân bản là gì? Lí do nhân bản? Nhân bản không hiệu quả với lệnh gì?**

* Lý do : truy xuất nhanh, tính sẵn sàng cao
* Không hiệu quả: update, delete, insert

**Câu 3: Phân hoạch là gì?**

* Phân hoạch là dữ liệu không nhân bản và nó chỉ tồn tại một bản sao duy nhất cho mỗi phân mảnh trên mạng.

**Câu 4: Nêu rõ các quy tắc phân mảnh: Tính tái thiết, tính đầy đủ, tính tất biệt**

* Tính tái thiết: Một quan hệ R được phân mảnh thành các quan hệ R1, R2, … Rn , thì cần định nghĩa phép toán sao cho tái thiết lai quan hệ gốc được.
* Tính đầy đủ: Nếu một quan hệ được phân rã thành các mảnh R1, R2,.., Rn thì mỗi mục dữ liệu trong R cũng có thể có trong một hay nhiều Ri.(đặc tính này nói lên sự phân mảnh không mất thông tin)
* Tính tách biệt: Nếu một quan hệ R được phân rã thành các phân mảnh R1, R2, …Rn và mục dữ liệu di nằm trong phân mảnh Rj sẽ khong nằmt trong bất kì một phân mảnh Rk nào khác.

**Câu 5: Sp trong suốt là gì? Điều kiện để trong suốt? Có mấy mức trong suốt, nêu rõ các mức trong suốt đó?**

* SP trong suốt là sp thực hiện ở bất kì server nào cũng cho kết quả đúng như nhau.
* Điều kiện để trong suốt:
  + Cùng tên LINKSERVER
  + Cùng tên cơ sở dữ liệu
* Có 4 mức trong suốt:
  + Trong suốt phân tán: Dữ liệu phân tán xuống và tên các database phân tán giống nhau.
  + Trong suốt định vị: Khi gọi tên các SP phân mảnh này chỉ cần gọi tên mà không cần gọi ra vị trí bảng đó
  + Trong suốt ánh xạ cục bộ: phải cung cấp tênn các phân mảnh và vị trí cấp phát chúng.
  + Không trong suốt: phải viết lệnh theo hệ quản trị cơ sở dữ liệu tương ứng.

**Câu 6: LinkServer dùng để làm gì? Khi tạo LinkServer cần điều kiện gì? Muốn truy xuất đến phân mảnh gần nhất đầu tiên chúng ta phải tạo LinkServer như thế nào?**

* LinkServer: dùng để truy xuất dữ liệu các subcriber
* Điều kiện: LinkServer cùng tên link và cùng tên cơ sở dữ liêu.
* Phụ thuộc vào cách đặt tên link sao cho phù hợp (ví dụ: gần đặt 1, xa đặt 2)

**PHẦN IV: Tối ưu hóa truy vấn**

**Câu 1: Cách tối ưu hóa truy vấn**

* Thực hiện phép chọn, chiếu trước, phép kết sau. Tìm cách khử phép (nếu được). Nếu một điều kiện xuất hiện nhiều lần trong where thì dùng các biến đổi tương đương để làm cho điều kiện xuất hiện một lầ. Trong mệnh đề AND điều kiện nào có xác xuất sai cao đặt đầu, mệnh đề OR ngược lại.

**Phần V: Giao tác phân tán**

**Câu 1:** Giao tác phân tán là gì? Các loại giao tác? Sự giống nhau và khác nhau giữa hai giao tác này? Sự giống nhau và khác nhau của giao tác phân tán và giao tác tập trung? Khi viết cần bật dịch vụ gì? Cách bật như thế nào?

* Giao tác phân tán là tập các lệnh đọc ghi và các bước tính toán cần thiết lên cơ sở dữ liệu và có thêm khả năng tái thiết dữ liệu( nếu không có khả năng tái thiết thì giao tác phân tán là một SP thuần túy).
* Các loại giao tác: Giao tác phẳng (chỉ có một lệnh BEGIN TRANSACTION và một lệnh COMMIT TRACSACTION) và giao tác lồng(có thể có nhiều lệnh BEGIN và COMMIT lồng vào nhau).
* Sự khác nhau giữa giao tác phân tán và giao tác tập trung

+ Giao tác phân tán: Giao tác phân tán phân bố nhiều nơi. Khi hai giao tác chạy ở hai server thì nếu một giao tác gặp sự cố thì giao tác kia vẫn chạy bình thường.

+ Giao tác tập trung: Giao tác tập trung ở một nơi. Khi giaoc tác gặp sự cố thì tất cả các giao tác khác đều dừng.

* Khi viết giao tác cần bật dịch vụ DISTRIBUTE TRANSATION COODINATOR
* Vào run gõ services.msc->start Distribute Transaction Coodinator.

**Câu 2: Nêu rõ các tính chất giao tác: nguyên tử, nhất quán, biệt lập, bên vững? ý nghĩa của các tính chất?**

* Tính nguyên tử: Một giao tác được xử lý như một đơn vị hoạt tác. Chính vì thế nếu một lệnh trong giao tác bị lỗi thì tất cả các lẹnh trong giao tác sẽ bị hủy bỏ. và ngược lại
* Tính nhất quán: khi thực hiện các lệnh insert, update, delete nếu thành công thì tất cả dữ liệu phải được cập nhật hết,còn nếu bị lỗi thì dữ liệu phải trở về vị trí ban đầu.
* Tính biệt lập: Khi một thao tác đang thực thi thì các giao tác khác không thể thấy dữ liệu của nó.

Ví đang delete thì không thể thêm đợi delete xong mới được thêm.

* Tính bền vững: một giao tác khi được commit thì kết quả của nó vẫn được duy trì có định và không bị xóa ra khỏi csdl.

**Câu 3: Giải thích cách lệnh BEGIN TRANSACTION, COMMIT TRANSACTION, @@TRANCOUNT, DEADLOCK, ROLLBACK, ABORT?**

* @@TRANCOUT: số giao tác đang được thực hiện trên giao tác hiện tại.
* BEGIN TRANSACTION: bắt đầu một thao tác, tăng @@TRANCOUNT thêm 1
* COMMIT TRANSACTION: giảm 1 trên @@TRANCOUNT
* ROLLBACK: đưa @@TRANCOUNT về 0

**Câu 4: Dịch vụ MS Distributed Transaction Coordinator là gì?**

* Là một trình quản lí điều phối các giao tác phân tán, nó cho phép các ứng dụng của client thao tác lên dữ liệu của các datasource trong một giao tác phân tác.
* MS DTC service điều phối sự đúng đắn của một giao tác phân tán, nó bảo đảm rằng hoặc là tất cả được cập nhật trên các server là được thực hiện, hoặc trong trường hợp lỗi thif xem như chưa thực hiện giao tác gì trên giao tác đó.
* Lưu ý: Nếu MSDCT không hoạt động thì cách khắc phục như sau:
  + Vào cmd gõ msdtc.exe-install
  + Sau đó start service Distribute Distributed Coordinator

Nếu không khởi động được thì làm một trong hai cách sau:

* + - Cách 1: Re-create msdtc log for DTC service
      * Vào cmd gõ msdtc -resetlog
      * Gõ tiếp net start msdtc
    - Cách 2: set the logon account permisstion for msdtc

**Phần VI : FORM**

**Câu 1: Các thuộc tính của ComboBox là gì? Ý nghĩa của nó?**

* BindingSource: chỉ ra dữ liệu mà ComboBox đó nằm ở đâu.
* DisplayMemBer: Dữ liệu hiện thị để người dùng chọn. Dữ liệu có thể là một cột trong DataTable
* ValueMember: là giá trị tương ứng với displaymember. Dữ liệu có thể là một cột trong DataTable

**Câu 2: Tại sao đăng nhập ở form đăng nhập thì mình có thể đăng nhập thành công được? khi đăng nhập thành công , tại sao biết được mã, họ tên , nhóm của người đăng nhập**

**Câu 3: “Device” trong BACKUP DATABASE QLVR TO DEVICE\_QLVT là gì?**

* Device là tên thiết bị dùng để lưu trữ database khi backup và restore, đường dẫn: tên server /Server Object/Backup Device/DEVICE\_QLVT.

Đặt tên có Device để phân biệt đây là 1 colum, không phải là một DB

**Câu 4: Khi backup db, thì bản sao lưu đó sẽ được sẽ được lưu trữ ở đâu?**

* Được lưu ở table backupset, đường dẫn tên server.Databases/SystemDatabase/msdb/Tables/System Table/backupset
* Có 2 dạng backup là update là override
  + Dạng UPDATE: backup bản này nối tiếp bản kia
  + Dạng OVERIDE: khởi tạo lại bản backup

**Câu 5: Client và server liên kết, trao đổi dữ liệu bằng cách nào?**

* Thông qua DataAdapter

**Câu 6: Nút lệnh thêm, cái gì gọi AddNew();**

* BindingSource

**Câu 7: Để update dữ liệu và db, trước khi gọi lệnh Update thì cần phỉa thực hiện lệnh gì?**

* EndEdit();
* ResetCurrentItem();

**Câu 8: DataSet, DataTable, DataAdapter là gì?**

* DataSet có thể hiểu là một bản ghi tạm, nó lưu trữ và chỉnh sửa dữ liệu ở Local Cache, tức là khi có ngắt kết nối tớ server đi nữa thì vẫn thao tác với dữ liệu trên DataSet. Sau khi xem sét và chính sửa dữ liệu xong ta tạo kết nối (DataAdapter) để Update dữ liệu từ Local(DataSet) về DataSource(SQL)
* Dữ liệu trong DataSet được lưu ở dạng DatâTble(DT). DT là một đối tượng chứa dữ lệu từ DataBase gởi qua, cấu trức gồm có các DataRow và DataColumn. DT có thể là 1 Table, 1 View, hoặc 1 SP, chứa dữ lệu nhưung không chứa lệnh.
* DataAdapter(DA) là một dạng cầu nối, giúp cácc lệnh như Select, Insert, Update, Delete để trao đỏi dữ liệu giữa Client và Server, và DA đổ dữ liệu từ DataSource về DS bằng phương thức Fill, GetData

**Câu 9: BindingSource là gì?**

* Là nơi chỉ ra (Binding) để thao tác dữ liệu, là cầu nối giưuã DataGridView và DataTable.

**Câu 10: Các lệnh EndEdit, ResetCurrentItem sử dụng trong bài có ý nghĩa gì?**

* AddNew() thêm một item/row vào danhh sách,
* CancelEdit() bỏ qua các dữ liệu đang chỉnh sửa trong rrơ và tả con trỏ về row cuối cùng. Chỉ có tác dụng với lệnh AddNew và dữ liệu đang được hiệu chỉnh nhưng chưa ghi vào DataSet.
* EndEdit: kết thúc quá trình chỉnh sửa dữ liệu và gơi và DataSet.
* ResetCurrentItem: Reread/ đọc lại row hiện tại ta đang đứng, dữ liệu hiện tại ở row ta đang đưungs không còn ở dạng tạm nữa(tức là dữ liệu có thể cancle edit) mà hiện thị dữ liệu đã chính thức ghi vào DataSet
* Update cập nhật data từ DataSet về DB thông qua TableAdapter.
* Count(): đếm xem số record tương ứng với row đang chọn tồn tại bao nhiêu bảng trong bindingsource được kiểm tra.
* RemoveCurrent Xóa row đang chọn ra khỏi DataSet
* **3 quy tắc phân mảnh**
* + Tính đầy đủ: dữ liệu nằm ít nhất ở 1 mảnh
* + Tính tái thiết được: thỏa tính đầy đủ thì thỏa tính tái thiết
* Định nghĩa 1 phép toán để tái thiết dữ liệu, ngang là phép hội, dọc là phép kết tự nhiên trên khóa chính
* + Tính tách biệt: dữ liệu chỉ nằm duy nhất ở 1 mảnh(tính chất này có thể vi phàm được, table nào nhân bản thì vi phạm)
* **4 Tính chất của giao tác**
* **Tính nguyên tử**: 1 tập lệnh xem như 1 lệnh, hoặc thực hiện được hết hoặc không lệnh nào được thực hiện ( đây là tính chất quan trong nhất của giao tác)
* **Tính nhất quán**: sau khi xử lý thì chuyển từ trạng thái nhất quán này -> trạng thái nhất quán khác (**Có 3 độ trang 82)**
* **Tính bền vững**: sau khi ủy thác thì dữ liệu không bị xóa khỏi database-> để khôi phục dữ liệu
* **Tính biệt lập**: khi một giao tác đang thực thi không lộ ra các kết quả của nó cho những giao tác khác đang cùng hoạt động trước khi nó ủy thác.
* read uncommitted
* read committed
* repeatable read
* Serialitabe
* snapshot:
* **NHỮNG CÁCH TỐI ƯU HÓA TRUY VẤN**
* Chọn chiếu trước, kết sau
* Khử phép kết nếu được
* Ví dụ cụ thể :
* Khi chưa khử kết: lấy ra tên sv và tên lớp của sinh viên thuộc lớp CN04 : select SINHVIEN.TEN, LOP.TEOLOP from SINHVIEN join LOP on SINHVIEN.MALOP=LOP.MALOP where LOP.MALOP = “CN04”
* Khi đã khử kết: dùng truy vấn lồng :select TEN (select TENLOP from LOP where MALOP=”CN04”) as tenlop from SINHVIEN where MALOP = “CN04”
* 1 điều kiện xuất hiện nhiều lần, thì sử dụng phép biến đổi tương đương
* MĐ And: đk có xác suất sai cao đặt ở đầu
* MĐ Or: đk có xác suất đúng cao đặt ở đầu
* Field tham gia trong mệnh đề quan hệ nên được sắp xếp thứ tự trước và thứ tự này phải được sử dụng trong mệnh đề truy vấn với mệnh đề WITH(Index\_Tên Index)
* **THẾ NÀO LÀ 1 SP TRONG SUỐT**
* Sp là trong suốt là 1 sp khi ta cho thực thi ở 1 server thì vẫn cho thực thi được ở những server còn lại mà không cần chỉ rõ đường dẫn đến table cần truy xuất (hay có thể trả lời : là sp mà khi ta cho thực thi ở bất kì phân mảnh nào đều cho kết quả giống nhau
* **Điều kiện để viết sp trong suốt**
* Link giống nhau,tên database giống nhau
* **Tại sao dùng remote login**
* **Link server**
* Mục đích **:**cho phép truy cập dữ liệu từ server này sang server khác
* Cách tạo: ví dụ tạo link ở server 1 để truy xuất dữ liệu ở server 2
* Đứng ở server 2 tạo login HOTROKETNOI
* Đứng ở server 1 tạo link
* Số link : n(n-1) n là số server phân mảnh
* Số tên link : n-1
* Cú pháp sử dụng : TENLINK.TENDATABASE.DBO.TENBANGCANTRUYXUAT
* **CÁC LOẠI PHÂN MẢNH** : NGANG DỌC, HỖN HỢP
* **Phân mảnh ngang nguyên thủy** là phân mảnh của 1 quan hệ dựa trên 1 vị từ được định nghĩa trên quan hệ đó
* VD TENKHOA=”VT”
* **Phân mảnh ngang dẫn xuất** là phân mảng của 1 qh trên vị từ được định nghĩa trên quan hệ khác.
* VD: Sinhvien.MALOP=Lop.MALOP
* **Phân mảnh dọc:** phân mảnh dựa trên khóa chính của một quan hệ ( phải có khóa chính để đảm bảo tính tái thiết)
* **Phân mảnh hỗn hợp** : vừa ngang vừa dọc
* **Điều kiện để phân tán được**: phải có dịch vụ SQL server Agent
* **Giao tác là gì:** 1 dãy các thao tác đọc ghi trên cơ sở dữ liệu cùng với các phép toán cần thiết: thêm, xóa, sửa có tính nhất quán để giải quyết các tình huống khi dữ liệu bị mất tính nhất quán khi có nhiều truy xuất đồng thời
* **Để viết giao tác cần phải bật dịch vụ**  MSDTC Miscrosoft Distribute Transaction Coordinator
* **Phân tán có 2 bước**
* *Định nghĩa publication***:** 1 containerchứa các article (table, view, stored procedure, udf) Trong đó phải chỉ rõ các server:
* Publisher chứa bản cơ sở dữ liệu gốc
* Distributor điều phối phân tán dữ liệu
* *Định nghĩa subcription( database)* 1 container nhận publication. Trong đó phải chỉ rõ server subcriber chứa cơ sở dữ liệu sau khi phân tán
* Trong 1 server  có nhiều user - ten user co the trung nhau
* Trong 1 database có nhiều user, tên khong dc trung nhau
* Trong 1 server có nhiều login, tên login không dc trùng nhau
* **Lệnh select...union** : dùng để lấy dữ liệu từ các server phân mảnh về
* Ví dụ : SELECT MALOP,TENLOP FROM DBO.LOP
* UNION SELECT MALOP,TENLOP FROM LINK1.QLDSV.DBO.LOP
* **Trong lược đồ phân mảnh, số lượng db trong publication và số lượng db trong subcriber như thế nào ?** Số lượng db trong publication nhiều hơn vì trong publication chứ nhieuf subcription
* **Trong db, cái nào là nhân bản, cái nào là phân hoạch** (mở cây dẫn xuất ra xem, cái nào không có trong cây dẫn xuất là nhân bản, cái đầu tiên là phân mảnh ngang nguyên thủy)
* **Nếu đã phân tán xong cơ sở dữ liệu, muốn thay đổi cấu trúc cột của server gốc thì làm ntn?** Có thể thay đổi bằng cách viết sp dùng lệnh ALTER TABLE ở server gốc sau đó đồng bộ xuống các server phân mảnh
* **Dữ liệu sau khi nhập form sẽ được đẩy về đâu** : đẩy về publisher sau đó đồng bộ xuống các subcriber
* **Trong các table, cái nào mạng tính đầy dủ, cái nào vi phạm tính tách biệt** : các table nhân bản thì vi phạm tính tách biệt, tất cả các table còn lại thì mang tính đầy đủ.
* **Giao tác với giao tác phân tán giống và khác nhau như thế nào?**
* Giao tác thì thực thi trên môi trường cơ sở dữ liệu tập trung (gồm có giao tác phẳng và giao tác lồng), còn giao tác phân tán thì thực thi trên môi trường cơ sở dữ liệu phân tán. Giống nhau thì nêu 4 tính chất giao tác.
* **Loginname nằm trong table nào?** Nằm trong table sys.sysuser trong DB đó
* **Tại sao biết user liên kết với login nào?** Username và loginname liên kết với nhau qua trường sid (trên user và login đều có sid) nên từ loginname biết được username từ sid
* **Hàm suy từ loginname -->username: SUSER.SID(ten login):**-->sid --> username --> userid --> groupid --> rolename
* **Tên nhóm quyền nằm trong table nào?** Sys.sysuser
* **Rowguid** : do hệ thống tự sinh ra, dùng để đồng bộ dữ liệu xuống phân mảnh, nó chỉ rõ table nào, cột nào để đưa dữ liệu xuống đúng vị trí đó
* **Ưu khuyết điểm của nhân bản:** truy xuất nhanh, đứng ở đâu cũng có thể select đc. Nhưng update chậm vì có quá nhiều bản sao
* **Ưu khuyết điểm của phân hoạch :** select chậm nhưng insert và update nhanh do chỉ thao thác trên server gốc hoặc server phân mảnh cần truy xuất.
* **Trong suốt phân tán, có bao nhiêu mức trong suốt, người lập trình đứng ở mức nào?** 4 mức, nhười lập trình đửng ở múc 3 nhưng làm cho người dùng cảm thấy như đang ở mức 1.
* Trong suốt phân mảnh: không cần chỉ rõ phân mảnh cũng như vị trí cấp phát cho phân mảnh
* Trong suốt vị trí : cung cấp tên phân mảnh nhưng không cần chỉ ra vị trí phân mảnh
* Trong suốt ánh xạ cục bộ : cung cấp tên phân mảnh và vị trí cấp phát
* Không trong suốt
* **Tại sao table đó nhân bản?**
* Trong đề vật tư thì vật tư nhân bản : vì vật tư có thể có ở chi nhánh này cũng có thể có ở chi nhánh khác
* Trong đề trắc nghiệm thì giáo viên nhân bản vì giáo viên có thể dạy ở cở sở này cũng có thể dạy ở cơ sở khác, bộ đề đi theo giáo viên nên cũng nhân bản. Giáo viên cở sở 1 có thể dạy ở cở sở 2 nhưng không thể đăng kí thi cho lớp ở cơ sở 2.

**-Giao tác là gì:** là 1 dãy các thao tác đọc ghi trên CSDL, cùng các phép toán: thêm, xóa,sửa có tính nhất quán để giải quyết các tình huống dữ liệu bị mất tính nhất quán khi có nhiều truy xuất đồng thời. +Để viết giao tác cần bật dịch vụ: MSDTC (Miscrosoft Distribute Transaction Coordinator )

-**Các loại giao tác** :Giao tác phẳng & giao tác lồng

+**Giao tác phẳng**: chỉ có 1 lệnh Begin Transaction & 1 lệnh Commit Transaction

+**Giao tác lồng**: có thể nhiều lệnh Begin Transaction & Commit Transaction lồng với nhau

- **Có 4 tính chất giao tác:** nguyên tử, nhất quán, bền vững, biệt lập.

+**Tính nguyên tử**: 1 tập lệnh được xem như là 1 lệnh, thực hiện được hết hoặc không thực hiện lệnh nào (tính chất quan trọng nhất của giao tác).

+**Tính nhất quán**: nói lên tính đúng đăn của dữ liệu, sau khi xử lý thì chuyển từ trạng thái nhất quán này sang trạng thái nhất quán khác ( có 3 độ )

[ T ko đè lên dữ liệu rác của ~ g/tác khác, T ko đọc dữ liệu rác của ~ g/tác #]

+**Tính bền vững**: sau khi ủy thác thì dữ liệu k bị xóa khỏi database ( để khôi phục dữ liệu)

+**Tính biệt lập**: khi 1 giao tác đang thực thi ko lộ ra các kết quả của nó cho ~ giao tác khác đang cùng hoạt động trước khi nó ủy thác.

+ Read Uncommited

+ Read Commited

+ Repeatable Read

+ Serializable

+ Snapshot

**- Sự khác nhau giữa giao tác phân tán tập trung & giao tác phân tán**

+Giao tác phân tán: phân bố ở nhiều nơi. Khi 2 giao tác chạy ở 2 server nếu 1 giao tác gặp sự cố thì giao tác kia vẫn chạy bình thường

+Giao tác tập trung: tập trung chỉ ở 1 nơi. Khi giao tác gặp sự cố thì tất cả thao tác đều dừng.

**Giao tác với giao tác phân tán giống và khác nhau ntn?** Giao tác thì thực thi trên môi trường CSDL tập trung ( giao tác phẳng, giao tác lồng), còn giao tác phân tán thì thực thi trên môi trường CSDLPT. Giống 4 t/chất giao tác

**Tính trong suốt của 1 SP? Điều kiện để biết 1 SP trong suốt?**

- SP trong suốt là SP khi chạy trên tất cả các phân mảnh phải cho ra các kết quả như nhau.

Điều kiện để viết 1 SP trong suốt là tên database ở các phân mảnh phải giống nhau, tên link server các server phải giống nhau

**Có bao nhiêu mức trong suốt và người lập trình đứng ở mức nào?**

4 mức, ng lập trình đứng ở mức 3 nhưng làm cho ng dùng cảm thấy ở mức 1

+Trong suốt phân mảnh: ko cần chỉ rõ phân mảnh và vị trí cấp phát cho phân mảnh

+ Trong suốt vị trí: cung cấp tên phân mảnh nhưng ko cần chỉ ra vị trí của phân mảnh.

+Trong suốt ánh xạ cục bộ: cung cấp tên phân mảnh và vị trí cấp phát

+ Không trong suốt

**Trong suốt:** khi ng sử dụng thực hiện truy vấn thì k có cảm giác dữ liệu đang phân tán hay tập trung

**3 quy tắc phân mảnh:**

+ Tính đầy đủ : dữ liệu nằm ít nhất ở 1 mảnh

+ Tính tái thiết đc: thỏa tính đầy đủ thì thỏa tính tái thiết, định nghĩa 1 phép toán để tái thiết dữ liệu: ngang là phép hội ( union) , dọc là phép kết tự nhiên trên khóa chính.(join-on)

+Tính tách biệt: dữ liệu chỉ nằm duy nhất ở 1 mảnh ( tính chất này có thể vi phạm đc, table nào nhân bản thì vi phạm)

1. **Phân mảnh ngang là gì? Phân mảnh dọc là gì? Phân mảnh ngang nguyên thủy là gì? (Khoa) Phân mảnh ngang dẫn xuất là gì?( giảng viên, lớp, sinh viên , điểm)**

-Phân mảnh ngang ( phép hội union) là chia ( phân rã) 1 quan hệ theo các bộ. Mỗi phân mảnh ngang là tập con của quan hệ, có 2 loại phân mảnh ngang:

+Phân mảnh ngang nguyên thủy: 1 q/hệ dựa trên 1 vị từ định nghĩa trên 1 quan hệ đó. Vd: TENKHOA = “VT”

+P/mảnh ngang dẫn xuất: một q/hệ dựa trên 1 vị từ định nghĩa trên 1 q/hệ #.

VD: SINHVIEN.MALOP = LOP.MALOP

Vd: Chương trình QLDSV phân mảnh ngang theo khoa nên:

Khoa MAKH = “CNTT” => Khoa là phân mảnh ngang nguyên thủy.

Lớp MAKH = “CNTT” => Lớp là phân mảnh ngang nguyên thủy.

KHOA.MAKH = LOP.MAKH => Lớp là phân mảnh ngang dẫn xuất.

-Phân mảnh dọc ( phép kết tự nhiên có khóa chính): phân mảnh dựa trên khóa chính của 1 quan hệ.

-Phân mảnh dọc ( phép kết tự nhiên có khóa chính): phân mảnh dựa trên khóa chính của 1 quan hệ.

**Điều kiện phân mảnh dọc :** **có khóa chính**

**Vì sao các phân mảnh dọc cần có khóa chính: Để tái thiết đc dữ liệu.**

-Phân mảnh hỗn hợp : vừa ngang vừa dọc

**Điều kiện để phân tán được:** phải có dịch vụ SQL Server Agent.

**12. Khi nào cần phải tạo link server? Khi tạo cần các ràng buộc gì?**

-Tạo link server khi database nằm ở 2 server khác nhau , ràng buộc là: tên các server link tới nhau thì phải cùng tên.

Mục đích: cho phép truy cập dữ liệu từ server này sang server khác

-Các tạo: ví dụ tạo link ở server 1 để truy xuất dữ liệu ở server 2

- Đứng ở server 2 tạo login HTKN, đứng ở server 1 tạo LINK

- Số link : n(n-1) ; n là số server phân mảnh, số tên link : n-1

Cú pháp: TENLINK.TENDATABASE.DBO.TENBANGTRUYXUAT

**Các bước phân tán? Khi phân tán cần bật dịch vụ gì?**

- Cấu hình các server:

Bước 1: cấu hình distribution

+ Publisher (server gốc) : chứa bản cơ sở dữ liệu gốc

+ Distributor: điều phối các phân mảnh phân tán

+ Subscriber: server chứa cơ sở dữ liệu sau khi phân tán

Bước 2: Tạo publication

* Định nghĩa Publication: 1 container chứa actical ( view, sp, table, UDF), chứa lược đồ phân mảnh , thể hiện mối quan hệ dữ liệu giữa các table khi chúng ta phân tán dữ liệu.
* Định nghĩa Subcrition(database): 1 container nhận Publication

(chỉ định cái server nào sẽ chứa cái publication, chứa dữ liệu phân tán theo lược đồ phân mảnh)

Khi phân tán cần bật dịch vụ: SQL Server Agent

KHOA: phân mảnh ngang nguyên thủy,còn lại là dẫn xuất ,

Môn học: là nhân bản, vì môn học có thể có ở khoa này cũng có thể có ở khoa khác

(Continuosly : tự động đồng bộ hóa )

+ UDF: hàm do người dùng tự định nghĩa

-Trong 1 server có nhiều user – tên user có thể trùng nhau

-Trong 1 database có nhiều user – tên user ko đc trùng nhau

-Trong 1 server có nhiều login- tên login ko đc trùng nhau

-1 login chỉ thuộc 1 user

1. **Thuộc tính của folder RELPDATA là gì? ở đâu? Mục đích của folder này dùng để làm gì?**

* Thuộc tính của folder RELDATA là share, có quyền read/write. Mục đích của folder là để chứa các dữ liệu trao đổi trong quá trình update dữ liệu từ các phân mảnh về CSDL gốc, và từ CSDL gốc đến các phân mảnh.

**2. Trong lược đồ phân mảnh, số lượng database trong publication và số lượng database trong subscriber như thế nào?**

**-**Số lượng database trong publication nhiều hơn trong subcrition vì trong publication chứa nhiều subcrition.

**3. Khi chọn nào mà khoa đó xuất hiện, sự kiện gì làm điều đó? Lệnh nạp dữ liệu từ database về các datatable là gì?** (vd khi nhập điểm thì lệnh gì nạp về): Sai đồ men ( lệnh truy vấn lại) -----**Sử dụng servername**-----

**4. Trong database cái nào là nhân bản, cái nào là phân hoạch?**

**-**Môn học là nhân bản, còn lại là phân hoạch.

Loginname nằm trong table nào: Table sys.sysuser trong database đó.

**5. Tại sao khi gõ login lại biết họ tên?**

-Tại vì login và user liên hệ với nhau qua trường sid trong sys.systemlogin và sys.systemuser

**Tại sao biết user liên kết với login nào?** Username & loginname liên kết với nhau qua trường sid ( trên user và login đều có sid) nên từ loginname biết được username từ sid.

**Hàm suy từ loginname => username**: SUSER.SID (tên login) => sid => username => userid => groupid => rolename.

**Tên nhóm quyền nằm trong table nào?** Sys.sysuser

**Rowguid:** là do hệ thống tự sinh ra, dùng để đồng bộ dữ liệu xuống phân mảnh, nó chỉ rõ table nào, cột nào để đưa dữ liệu xuống đúng vị trí đó.

**Ưu khuyết điểm của nhân bản**: truy xuất nhanh, đứng ở đâu cũng có thể select ( chiếu) được. Nhưng update chậm vì có quá nhiều bản sao.

**Ưu khuyết điểm của phân hoạch**: select chậm nhưng insert và update nhanh do chỉ thao tác trên server gốc hoặc server phân mảnh cần truy xuất.

**6. Nếu phân tán xong CSDL, muốn thay đổi cấu trúc(cột) của table server gốc thì ta làm thế nào?**

**-**Có thể thay đổi cột bằng cách viết SP dùng lệnh ALTER TABLE ở server gốc sau khi thêm xong thì đẩy xuống phân mảnh con.

**7. Chuyển chữ thường thành chữ hoa** : dùng lệnh **upper**

**8.Đưa ra 2 cách xử lý truy vấn dữ liệu cùng lúc ở 2 server khác nhau?**

-Dùng câu truy vấn song song cho nó chạy song song ở 2 server khác nhau.

**9.Dữ liệu sau khi nhập ở các form sẽ đc đẩy vể đâu?**

-Đẩy về server publicsher sau đó đồng bộ xuống các Subcriber

**10. Trong các table cái nào là tách biệt? cái nào là đầy đủ?**

-Tất cả các table đều mang tính chất đầy đủ, các table nhân bản thì vị phạm tính tách biệt.

**11. Subform là gi?**

**-**Là form có ít nhất 2 bảng và có sự liên kết với nhau.

**14. Quy trình thực hiện tối ưu hóa truy vấn trong CSDLPT?**

-Chọn trước, chiếu trước , kết sau. Khử phép kết nếu được, khử kết bằng cách thay phép kết bằng phép truy vấn lồng.

+Tối ưu hóa các phép điều kiện bằng phép biến đổi tương đương

+ Chiếu : Select , chọn :from , điều kiện lọc : where , kết : join- on – where ( đkiện kết)

**18. Câu lệnh bắt đầu & kết thúc của 1 giao tác là gi?**

Begin distributed Transaction >> bắt đầu

Commit transaction >> Kết thúc

Bắt lỗi : Try (commit transaction) catch (Roliback transaction)

-**PGV**: đc quyền xem cả 2 khoa, và chỉ đc: một tài khoản duy nhất ở 1 phân mảnh nhưng đc quyền xem 2 phân mảnh (PGV)

**- Public static SqlConnection conn = new SqlConnection();**

Biến này dùng để chạy các SP và View, trong SQL nó như là cầu nối và chạy xuyên suốt trong chương trình.

**-ConnectionString**: thuộc tính quan trọng nhất vì chỉ khi mở kết nối kiểm tra xem chuỗi đó đúng hay sai

-**Lệnh Fill** : là lệnh tự động tải dữ liệu về

**-DropDownList:** trong cmb chỉ cho phép chọn, k đc ghi

**-Liên kết dữ liệu giữa 2 form**: biến thì nằm ở program, mà khi chạy thì nó nằm ở đăng nhập.

**-SelectedIndexChanged**: biến cố này tự động chạy khi ta chọn 1 giá trị mới trong cmb và khi nó chạy lấy ra servername nằm trong program.

-**DataSet**: là 1 đối tượng có thể chứa nhiều DataTable.

-**DataTable** dùng để lưu trữ 1 tập dữ liệu dạng bảng đc trích ra từ DataSource

-**DataTable khác với Table:** (1sp là 1 dataTable) là 1 đối tượng mà dữ liệu của nó là 1 phần của table.

**- Rowguid** : Tự sinh ra để đồng bộ hóa dữ liệu.

1. Có mấy loại Join: 3 loại

+ INNER JOIN

+ OUTER JOIN : Left Join & Right Join

+ CROSS JOIN: ko có khóa nó gộp lại nhưng vẫn lấy

**INNER JOIN** trả về các bản ghi có giá trị phù hợp giữa hai bảng

**LEFT JOIN** mọi bản ghi bảng bên trái trả về kết hợp với bản ghi phù hợp với bên phải nếu có (nếu không có thì nhận NULL)

**RIGHT JOIN** mọi bản ghi bảng bên phải trả về có kết hợp với giá trị phù hợp nếu có ở bảng trái

**OUTER JOIN** (full join) mọi bản ghi ở bảng trái và bảng phải kết hợp lại

-Nhân bản là gi? ở server nào cũng có dữ liệu như nhau .Lí do nhân bản? Nhân bản ko hiệu quả với lệnh gi?

- Các hàm lấy dữ liệu: dataSet và show dữ liệu: datagridview

-Lệnh select…union : dùng để lấy dữ liệu từ các server phân mảnh về

Vd: SELECT MALOP,TENLOP FROM DBO.LOP

UNION SELECT MALOP,TENLOP FROM LINK.QLDSV.DBO.LOP

-Đơn giản hóa 1 b/thức đại số quan hệ được t/hiện dựa trên các tiêu chuẩn :

+Tiêu chuẩn 1: Sử dụng phép biến đổi tương đương, phép chọn và chiếu

+Tiêu chuẩn 2:Thực hiện các phép càng xa càng tốt, tức là đẩy các phép chọn và các phép chiếu xuống phía dưới càng xa càng tốt.

+Tiêu chuẩn 3: Khi các phép chọn đc thực hiện sau 1 phép tích thì kết hợp các phép toán này để tạo thành 1 phép kết.

+Tiêu chuẩn 4: Kết hợp chuỗi các phép toán 1 ngôi liên tiếp nhau áp dungj cho 1 quan hệ toán hạng

+Tiêu chuẩn 5: Khi phát hiện các biểu thức con truy vấn, áp dụng các phép biến đổi tương đương để đơn giản hóa biểu thức truy vấn

* Truy vấn (query) là 1 biểu thức đc biểu diễn bằng 1 ngôn ngữ và dùng để xác định 1 phần dữ liệu đc chứa trong CSDL.
* Phân quyền:

Server 1+2: PGV : datareader, writer, securityadmin,access admin

KHOA: datareader, writer

USER: datareader, writer

Server 3: PKTOAN: accessadmin, reader, writer, securityadmin

-db\_accessadmin: ng dùng có quyền quản lý tài khoản SQL server đăng nhập

-db\_securityadmin: ng dùng có thể chỉnh sửa vai trò (role) và quản lý các bậc quản lý phân quyền khác

+ muốn tạo tài khoản thì dùng accessadmin + securityadmin ( cấp quyền)

-db\_datareader: ng dùng có thể đọc được toàn bộ dữ liệu.

-db\_datawrite: ng dùng có quyền thêm, xóa, sửa dữ liệu trong bảng

-db\_owner: toàn bộ ng dùng có quyền full-access

-sysAdmin: toàn bộ ng dùng đều có thể t/hiện các thao tác trên server.

**-Có 3 loại server**

+Local server: cài máy nào thì chạy máy đó

+Remote server: điều khiển từ xa giữa 2 máy

+ Link server : Liên kết server này với server khác.

-Dùng link cơ chế nối từ sql server đến database của server khác SQL server hay hệ thống khác

-Tạo link thì tạo remotelogin(HTKN) trước

1. **DeadLock là gì ? Các xử lý?**

- Một cv B gọi A: Nhưng A bị lỗi nên cv B cứ đợi A-> gọi là DeadLock

- Xử lý: thiết lập Timeout , sau bao lâu đó thì trở về trạng thái ban đầu

**Vị từ là gì ?** Thực chất là phép chọn trên 1 feild trên 1 giá trị const

1. **Dữ liệu rác là gì** ? Là dữ liệu chưa được ủy thác

**0. Để phân tán cơ sở dữ liệu** có 3 bước

Bước 1: định nghĩa distributor

+) Publisher chứa cơ sở dữ liệu gốc

+) Distribor chứa dữ liệu trung gian giữa cơ sở dữ liệu gốc và các phân mảnh

+) Subcriber chứa cơ sở dữ liệu sau khi phân tán

Bước 2: định nghĩa publication chứa các article ( sp, view, table )

Bước 3: định nghĩa subcrition chứa publication

**1.Có 3 hình thức phân mảnh** là phân mảnh ngang , dọc & hỗn hợp

2**.Phân mảnh ngang** là chia một quan hệ thành các bộ. Mỗi bộ là tập con của quan hệ.

Phân mảnh ngang có 2 kiểu gồm

+) Phân mảnh ngang nguyên thủy là phân mảnh dựa trên một vị từ được định nghĩa từ chính quan hệ đó.VD: MACN='CN1'

+) Phân mảnh ngang dẫn xuất là phân mảnh dựa trên một vị từ được định nghĩa dựa trên quan hệ khác.VD: NHANVIEN.MaKho = KHOHANG.MaKho

**3.Phân mảnh dọc** là dựa trên quan hệ kèm theo khóa chính ( bắt buộc phải có khóa chính để bảo toàn tính tái thiết)

Phân mảnh hỗn hợp là kết hợp của phân mảnh ngang & phân mảnh dọc

**4. Có 3 quy tắc phân mảnh**

- Tính đầy đủ: mỗi mục dữ liệu nằm trên ít nhất một phân mảnh.

- Tính tái thiết: khi một cơ sở dữ liệu chia thành nhiều phân mảnh thì khi hợp các phân mảnh lại ta sẽ có phân mảnh gốc.

- Tính tách biệt: mỗi mục dữ liệu chỉ nằm ở một phân mảnh duy nhất

Tính tách biệt có thể bị vi phạm nếu đó là nhân bản

Để phân tán thì cần có dịch vụ SQL Agent.

**5. Khi xây dựng cơ sở dữ liệu phân tán phải đảm bảo 2 tiêu chí.**

+) Tiêu chí cực tiểu: một phân mảnh khi sinh ra thì có ít nhất một ứng dụng sẽ sử dụng tới.

+) Tiêu chí đầy đủ: xác suất để một ứng dụng sử dụng tới các phân mảnh là như nhau.

**6. Sự trong suốt phân tán** là khi một sp chạy ở bất kì phân mảnh nào cũng cho kết quả tương tự. Người dùng không có cảm giác là đang

thao tác trên môi trường phân tán

Điều này đạt được khi có LINK Server giống nhau & tên database giống nhau.

**7. Có 4 mức trong suốt phân tán**

+) Trong suốt phân tán : không cần chỉ ra đường dẫn, nơi cấp phát phân mảnh. Người dùng cảm giác như đang thao tác trên cơ sở

dữ liệu tập trung

+) Trong suốt vị trí: cần chỉ rõ tên table cần truy vấn.

+) Trong suốt ánh xạ cục bộ: cần chỉ rõ tên phân mảnh và vị trí cấp phát

+) Không trong suốt: cần phải chỉ rõ tên database cần thao tác

Người lập trình đứng ở mức 3 & làm sao cho tạo ra mức 1.

8. **Giao tác** là một chuỗi các hành động đọc & ghi dữ liệu, kết hợp với các bước tính toán nhất quán ( begin transaction, rollback,

begin distributed transaction , commit) để xử lý các trường hợp dữ liệu bị thiếu sự nhất quán khi có nhiều truy vấn đồng thời.

9. **Có 2 loại giao tác chính** là giao tác tập trung & phân tán

+) Giao tác tập trung là giao tác hoạt động trên môi trường tập trung.Nếu xảy ra sự cố thì mọi hành động đều dừng

Gồm 2 loại là

(\*) Giao tác phẳng: có điểm khởi đầu duy nhất ( begin transaction ) và một điểm kết thúc duy nhất ( end transaction )

(\*) Giao tác lồng: chứa nhiều giao tác với điểm khởi đầu và duy nhất riêng biệt.

+) Giao tác phân tán hoạt động trên môi trường phân tán.Chúng có thể chạy đồng thời ở các phân mảnh được.

Nếu 1 trong 2 cái xảy ra sự cố thì cái còn lại vẫn hoạt động bình thường. Chỉ có một loại là giao tác phẳng

**10. Giao tác có 4 tính chất**

+) Tính nguyên tử: mọi lệnh sẽ được thực hiện thành công hoặc nếu một lệnh thất bại thì tất cả thất bại

+) Tính nhất quán: mọi rằng buộc được bảo toàn. Tính nhất quán được giữ nguyên.

+) Tính biệt lập: khi một giao tác đang hoạt động thì nó sẽ không để lộ kết quả cho giao tác khác.

+) Tính bền vững: khi một giao tác kết thúc thì kết quả của nó không thể bị đảo ngược.

11. Để sử dụng được giao tác phân tán cần có dịch vụ MS DTC - MicroSoft Distributed Transaction Cordinator.

**12. Có 5 cách để tối ưu hóa truy vấn**

+) Chọn trước, chiếu sau

+) Khử phép kết

+) Mệnh đề AND nào sai thì để ở đầu. Mệnh đề OR thì để ở cuối

+) Một điều kiện WHERE xuất hiện nhiều lần thì dùng các phép biến đổi tương đương để rút gọn lại

+) Field tham gia trong mệnh đề quan hệ nên được sắp xếp thứ tự trước và

thứ tự này phải được sử dụng trong mệnh đề truy vấn với mệnh đề WITH(Index\_Tên Index)

**13.Các mức cô lập dữ liệu**

- Read Uncommitted - mức yếu nhất - TT1 đang sửa đổi & TT2 có thể vào xem dữ liệu ngay cả khi TT1 chưa lưu thay đổi lại. Nói nôm na là “tôi không cần biết dữ liệu có đang được cập nhật hay không, hãy cho tôi dữ liệu hiện có ngay tại thời điểm này”.

- Read Commited - mức mặc định - TT1 đang sửa đổi dữ liệu thì TT2 không thể xem dữ liệu này. Chỉ khi TT1 xong thì TT2 mới xem được dữ liệu. Nếu TT2 sửa dữ liệu thì TT1 sẽ có được dữ liệu được sửa đổi từ TT2.

- Repeatable Read - mức an toàn - TT2 đang đọc một bảng dữ liệu thì TT1 không thể chỉnh sửa bảng dữ liệu này cho đến khi TT2 hoàn tất việc đọc dữ liệu đó. Nói nôm na là dữ liệu đang được đọc sẽ được bảo vệ khỏi cập nhật bởi các transaction khác

- Serializable - mức an toàn cao - TT1 đang sửa đổi một bảng dữ liệu thì TT2 không thể làm bất cứ hành động gì với bảng dữ liệu đó như: INSERT, UPDATE, DELETE,.....

- Snapshot - mức an toàn cao nhất - hoạt động tương tự Serializable, TT1 vẫn hoạt động, TT2 sẽ sửa dữ liệu trên một bản ghi đã được Snapshot sao chép.