**MỤC LỤC**

[**PHẦN I: PHÂN TÁN** 3](#_Toc104296670)

[1. Phân tán là gì? 3](#_Toc104296671)

[2. Đề tài QLVT thì chỗ nào là làm ứng dụng phân tán? (ví dụ chuyển tiền từ TK này đến TK khác - ứng dụng này sẽ yêu cầu cập nhật dl ở cả 2 nhánh) 4](#_Toc104296672)

[3. Kiến trúc của 1 hệ quản trị csdl phân tán 4](#_Toc104296673)

[4. Tại sao cần có cơ sở dữ liệu phân tán? 4](#_Toc104296674)

[5. So sánh ưu và nhược điểm của việc phân tán dữ liệu? 4](#_Toc104296675)

[6. Các bước phân tán? Khi phân tán cần bật dịch vụ gì? Cách bật như thế nào? 5](#_Toc104296676)

[7. Thuộc tính folder RELPDATA là gì? ở đâu? Mục đích của thư mục để làm gì? 6](#_Toc104296677)

[8. Trong lược đồ phân mảnh, số lượng database trong publication và số lượng database trong subscriber như thế nào? 6](#_Toc104296678)

[9. Có mấy loại server: 6](#_Toc104296679)

[**PHẦN II: SỰ TRONG SUỐT PHÂN TÁN** 6](#_Toc104296680)

[Câu 0: Cho biết ưu điểm và khuyết điểm của nhân bản và phân hoạch 6](#_Toc104296681)

[Câu 1: Các hình thức phân mảnh? (HÌNH THỨC # QUY TẮC # TIÊU CHÍ) 7](#_Toc104296682)

[Câu 1.5: Tiêu chí khi phân mảnh: 8](#_Toc104296683)

[Câu 2: Nhân bản là gì? Lí do nhân bản? Nhân bản không hiệu quả với lệnh gì? Ưu và khuyết điểm của nhân bản 8](#_Toc104296684)

[Câu 3: Phân hoạch là gì? 8](#_Toc104296685)

[Câu 4: Nêu rõ các quy tắc phân mảnh 8](#_Toc104296686)

[Câu 5: Sp trong suốt là gì? Điều kiện để trong suốt? Có mấy mức trong suốt, nêu rõ các mức trong suốt đó? 9](#_Toc104296687)

[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? 10](#_Toc104296688)

[Câu 7: Tại sao ngta sẽ phân tán những đơn đặt hàng (phiếu nhập/xuất) ko theo nhân viên mà phải theo kho? 10](#_Toc104296689)

[Câu 8: So sánh biến ảo cục bộ và biến ảo toàn cục 10](#_Toc104296690)

[Câu 9: Vị từ thích hợp là gì? 11](#_Toc104296691)

[Câu 10: Vị từ là gì? Một vị từ đơn giản là gì? Một vị từ sơ cấp là gì? 11](#_Toc104296692)

[Câu 11: Tiêu chí đầy đủ và tiêu chí cực tiểu là gì? 11](#_Toc104296693)

[**PHẦN III: THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN** 12](#_Toc104296694)

[**Các mục tiêu của việc thiết kế phân tán dữ liệu** 12](#_Toc104296695)

[**PHẦN IV: TỐI ƯU HÓA TRUY VẤN** 12](#_Toc104296696)

[Câu 1: Cách tối ưu hóa truy vấn 12](#_Toc104296697)

[**PHẦN V: GIAO TÁC PHÂN TÁN** 13](#_Toc104296698)

[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? 13](#_Toc104296699)

[Câu 3: Giải thích cách lệnh BEGIN TRANSACTION, COMMIT TRANSACTION, @@TRANCOUNT, DEADLOCK, ROLLBACK, ABORT? 14](#_Toc104296700)

[Câu 4: Dịch vụ MS Distributed Transaction Coordinator là gì? 15](#_Toc104296701)

[Câu 5: XACT\_ABORT là gì? Nó nhận được mấy giá trị? 15](#_Toc104296702)

[**PHẦN VI: FORM** 15](#_Toc104296703)

[Câu 1: Các thuộc tính của ComboBox là gì? Ý nghĩa của nó? 15](#_Toc104296704)

[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 15](#_Toc104296705)

[Câu 3: “Device” trong BACKUP DATABASE QLVR TO DEVICE\_QLVT là gì? 16](#_Toc104296706)

[Câu 4: Khi backup db, thì bản sao lưu đó sẽ được sẽ được lưu trữ ở đâu? 16](#_Toc104296707)

[Câu 5: Client và server liên kết, trao đổi dữ liệu bằng cách nào? 16](#_Toc104296708)

[Câu 6: Nút lệnh thêm, cái gì gọi AddNew(); 16](#_Toc104296709)

[Câu 7: Để update dữ liệu và db, trước khi gọi lệnh Update thì cần phỉa thực hiện lệnh gì? 17](#_Toc104296710)

[Câu 8: DataSet, DataTable, DataAdapter là gì? 17](#_Toc104296711)

[Câu 9: BindingSource là gì? 17](#_Toc104296712)

[Câu 10: Các lệnh EndEdit, ResetCurrentItem sử dụng trong bài có ý nghĩa gì? 17](#_Toc104296713)

[Câu 11: more attribute 18](#_Toc104296714)

[Câu 12: Có mấy loại JOIN 19](#_Toc104296715)

[**CÂU HỎI NGẮN** 19](#_Toc104296716)

[1. DeadLock là gì? Các xử lý? 19](#_Toc104296717)

[2. Dữ liệu rác là gì? 19](#_Toc104296718)

[3. Tại sao khi gõ login lại biết họ tên? / Tại sao biết user liên kết với login nào? 19](#_Toc104296719)

[4. Tên nhóm quyền nằm trong table nào? Sys.sysuser 20](#_Toc104296720)

[5. Loginname nằm trong table nào? 20](#_Toc104296721)

[6. Tên nhóm quyền nằm trong table nào? 20](#_Toc104296722)

[7. Nếu phân tán xong CSDL, muốn thay đổi cấu trúc(cột) của table server gốc thì ta làm thế nào? 20](#_Toc104296723)

[8. Chuyển chữ thường thành chữ hoa 20](#_Toc104296724)

[9. Đưa ra 2 cách xử lý truy vấn dữ liệu cùng lúc ở 2 server khác nhau? 20](#_Toc104296725)

[10. Dữ liệu sau khi nhập ở các form sẽ đc đẩy vể đâu? 20](#_Toc104296726)

[11. Trong các table cái nào là tách biệt? cái nào là đầy đủ? 21](#_Toc104296727)

[12. Subform là gi? 21](#_Toc104296728)

[13. Câu lệnh bắt đầu & kết thúc của 1 giao tác là gi? 21](#_Toc104296729)

[14. Dữ liệu sau khi nhập form sẽ được đẩy về đâu 21](#_Toc104296730)

[15. Tại sao dùng remote login 21](#_Toc104296731)

[17. Viết Stored Procedure ở đâu là hợp lý ? 21](#_Toc104296732)

[32. Dữ liệu sau khi nhập form sẽ được đẩy về đâu? 25](#_Toc104296733)

[54. 13.Các mức cô lập dữ liệu 26](#_Toc104296734)

# **PHẦN I: PHÂN TÁN**

## **Phân tán là gì?**

* là một tập hợp dữ liệu của một hệ thống thông tin nhưng được phân bố trên nhiều địa điểm (site) của một mạng máy tính.
* 2 khía cạnh quan trọng trong csdlpt là:

+ Sự phân tán: dữ liệu không lưu trữ trên cùng một địa điểm, phân biệt với csdl tập chung

+ Mối tương quan luận lý: không được vi phạm 2 ràng buộc là

* Ràng buộc cứng: role định nghĩa sẵn trên db (khóa chính, khóa ngoại, not null, …) -> db sẽ auto chạy theo những rang buộc cứng này
* Ràng buộc mềm: trigger (chạy auto kiểm tra ràng buộc) (nên mắc trigger ở 2 nơi là client và server để tránh được TH đưa dl sai về hệ thống -> mất thời gian chạy và time gửi)

## **Đề tài QLVT thì chỗ nào là làm ứng dụng phân tán? (ví dụ chuyển tiền từ TK này đến TK khác - ứng dụng này sẽ yêu cầu cập nhật dl ở cả 2 nhánh)**

**(ứng dụng toàn cục hay còn gọi là ứng dụng phân tán)**

* Chuyển chi nhánh
* Kiểm tra MaPN, MaPX đã tồn tại hay chưa?

## **Kiến trúc của 1 hệ quản trị csdl phân tán**

gồm 4 thành phần: quản trị db, truyền dl trên mạng, định nghĩa đc từ điển dl, phân tán (qtrg nhất)

## **Tại sao cần có cơ sở dữ liệu phân tán?**

Do dù cầu thực tế

## **So sánh ưu và nhược điểm của việc phân tán dữ liệu?**

* Ưu điểm:

+ Chia sẻ dữ liệu và điều khiển phân tán: Người sử dụng tại một vị trí này có

thể truy xuất dữ liệu (được phép) ở vị trí khác

+ Độ tin cậy và tính sẵn sàng:

* Người sử dụng tại một vị trí này có thể truy xuất dữ liệu (được phép) ở vị trí khác
* Sự cố tại một vị trí không ảnh hưởng đến hệ thống.

+ Tăng tốc độ xử lý truy vấn

* Nhược điểm:

+ Chi phí lớn

+ Khó phát hiện lỗi

+ Chi phi xử lý tăng

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

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, stored procedure, UDF).

Trong đó, phải chỉ rõ các server:

Publisher - chứa bản gốc cơ sở dữ liệu

Distributor - điều phối phân tán dữ liệu

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

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

## **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

## **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.

## **Có mấy loại 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.

# **PHẦN II: SỰ TRONG SUỐT PHÂN TÁN**

## **Câu 0: Cho biết ưu điểm và khuyết điểm của nhân bản và phân hoạch**

(1 quan hệ mà khi ta phân tán thông qua nhiều site sẽ có 2 k/n là nhân bản và phân hoạch)

|  |  |  |
| --- | --- | --- |
|  | **Nhân bản** | **Phân hoạch** |
| **Ưu điểm** | Khả năng hỗ trợ truy vấn rất cao (vì nằm ở nhiều site nên 1 site đóng, qua site khác vẫn tìm được) | Độ khả tín thấp (Do đang truy cứu trên site bị stop thì thua luôn, ko đi đâu được) |
| **Khuyết điểm** | Update dữ liệu lâu (Do khi khi update ở 1 site, site này lại đồng bộ về site chủ, site chủ lại tìm những site có nhân bản để update) | Update dữ liệu nhanh (vì chỉ update đúng 2 nơi là site chủ và site chứa nó) |

## **Câu 1: Các hình thức phân mảnh? (HÌNH THỨC # QUY TẮC # TIÊU CHÍ)**

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

* 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.

* 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

## **Câu 1.5: Tiêu chí khi phân mảnh:**

+) 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.

## **Câu 2: Nhân bản là gì? Lí do nhân bản? Nhân bản không hiệu quả với lệnh gì? Ưu và khuyết điểm của nhân bản**

* Lý do: truy xuất nhanh, tính sẵn sàng cao
* Không hiệu quả: update, delete, insert
* Ưu điểm: truy xuất nhanh, đứng ở đâu cũng có thể select ( chiếu) được. Nhưng update chậm vì có quá nhiều bản sao.
* Khuyết điểm: select chậm nhưng insert và update nhanh do chỉ thao tác trên server gốc hoặc server phân mảnh cần truy xuất.

## **Câu 3: Phân hoạch là gì?**

* Phân hoạch là dữ liệu không nhân bản và nó chỉ tồn tại một bản sao duy nhất (nằm ở site chủ và 1 site con)

## **Câu 4: Nêu rõ các 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. ngang là phép hội (union), dọc là phép kết tự nhiên trên khóa chính (join-on)

- 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**

**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

## **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 tương tự và người dùng cảm nhận như đang thao tác trên môi trường cơ sở dữ liệu tập chung
* Điều kiện để trong suốt:
  + Cùng tên LINKSERVER
  + Cùng tên cơ sở dữ liệu ở các subscriber phải giống nhau
* Có 4 mức trong suốt:
  + Mức cao nhất : Dữ liệu phân tán xuống và tên các database phân tán giống nhau.
  + Trong suốt vị 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 tên 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.

**Người lập trình đứng ở mức 3 và tạo ra mức 1**

## **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)
* Số link : n(n-1) ; n là số server phân mảnh,
* số tên link : n-1
* Cú pháp: TENLINK.TENDATABASE.DBO.TENBANGTRUYXUAT

## **Câu 7: Tại sao ngta sẽ phân tán những đơn đặt hàng (phiếu nhập/xuất) ko theo nhân viên mà phải theo kho?**

VÌ nhân viên we sẽ có 1 cái nghiệp vụ là điều khiển nvien đi từ chi nhánh này qua làm việc ở chi nhánh #. thì nếu mà làm lập trình như csdl tập chung mà sửa mã chi nhánh của nv từ CN1 -> CN2 thì hệ thống phân tán nó thấy rằng là nv này đang ở chi nhánh 1 sửa thành chi nhánh 2 thì auto cái mã đó sẽ nhảy qua chi nhánh 2. Nh mà khi nv đó nhảy qua chi nhánh 2 thì all những phân mảnh ngang dẫn xuất theo nv (vd đơn đặt hàng, phiếu xuất/nhập) sẽ auto di chuyển theo (theo cơ chế auto) => làm cho bài toán bị sai. VÌ all đơn đặt hàng, phiếu xuất/nhập của chi nhánh nào thì chỉ có chi nhánh đó thấy thôi, nếu như vậy thì sao mà tính lợi nhuận, doanh thu.

## **Câu 8: So sánh biến ảo cục bộ và biến ảo toàn cục**

|  |  |  |
| --- | --- | --- |
|  | **Biến cục bộ** | **Biến toàn cục** |
| **Giống** | Đều là table ảo. Thì cái table này nằm trong bộ nhớ trong nên sẽ auto mất đi khi người cấp cho nó thoát đi, đều biến mất | |
| **Khác** | Chỉ người tạo thấy cái table ảo mà thôi | Khi table này còn tồn tại trong bộ nhớ trong thì all các user # khi kết nối vào csdl đó thành công đều sẽ truy vấn đc dl trong table ảo toàn cục này |

## **Câu 9: Vị từ thích hợp là gì?**

Thỏa tính đầy đủ và tính cực tiểu

Và khi tạo ra 1 phân mảnh "thích hợp" thì phải có stored procedure sử dụng tới phân mảnh này.

## **Câu 10: Vị từ là gì? Một vị từ đơn giản là gì? Một vị từ sơ cấp là gì?**

* Vị từ là Mệnh đề login có nhiều điều kiện
* Một 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.

## **Câu 11: Tiêu chí đầy đủ và tiêu chí cực tiểu là gì?**

* Tiêu chí đầy đủ là ta có nhiều stored procedure thì xác suất nó truy xuất tới các phân mảnh là như nhau.
* Tiêu chí cực tiểu là mỗi site được sinh ra phải được sử dụng tới trong một stored procedure nào đó.

# **PHẦN III: THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN**

# **Các mục tiêu của việc thiết kế phân tán dữ liệu**

* Sự truy xuất cục bộ:

mục đích: dl trên từng mảnh là sẽ nhỏ => sẽ sinh ra 2 cái lợi:

+ dl càng ít truy vấn càng nhanh

+ che dấu đc dl của ta với các user khác trên cùng hệ thống

* Tính sẵn sàng và khả tín của các dữ liệu phân tán (còn phụ thuộc vào quan hệ đang là góc độ nhân bản 1 phần, hoàn toàn hay là phân hoạch)
* Sự phân bố tải (chắc chắn là đạt bởi vì khi ta phân tán tức là ta đang đẩy 1 phần dl sang 1 site và nguyên các qhe đó có hàng ngàn mẩu tin ta đẩy qua nhiều site và như vậy trên mỗi site dl nó sẽ bé hơn so với site gốc)
* Chi phí lưu trữ

# **PHẦN IV: TỐI ƯU HÓA TRUY VẤN**

## **Câu 1: Cách tối ưu hóa 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 thứ tự trước & thứ tự này phải đc sử dụng trong mệnh đề truy vấn với WITH (INDEX=ten\_index)

# **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** 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.
* **Các loại giao tác chính là**: giao tác tập trung & phân tán
* Giao tác tập trung: hoạt động trên môi trường tập chung, tập trung chỉ ở 1 nơi. Khi giao tác gặp sự cố thì tất cả thao tác đều dừng:

+**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

* Giao tác phân tán: hoạt động trên môi trường 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
* **Có 4 tính chất giao tác:** nguyên tử, nhất quán, bền vững, biệt lập.

+**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 chất quan trọng nhất của giao tác).

+**Tính nhất quán**: đảm bảo tất cả rằng buộc. Mọi lệnh đều được chạy

+**Tính biệt lập**: một 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**: một giao tác đã được commit thì dữ liệu đó được giữ nguyên và không thể rollback

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

+ Giao tác phân tán: hoạt động trên môi trường 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: hoạt động trên môi trường tập chung, tập trung chỉ ở 1 nơi. Khi giao tác gặp sự cố thì tất cả thao tác đều dừng.

* Giống: 4 t/chất giao tác
* Khi viết giao tác cần bật dịch vụ MS DTC (DISTRIBUTE TRANSATION COODINATOR)
* Vào run gõ services.msc->start Distribute Transaction Coodinator.

## **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?**

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

## **Câu 5: XACT\_ABORT là gì? Nó nhận được mấy giá trị?**

XACT\_ABORT: đây là tùy chọn kết nối. Nó chỉ nhận 2 giá trị là ON & OFF.

Nếu là OFF, SQL Server sẽ bỏ qua lệnh gây lỗi trong 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.

# **PHẦN VI: FORM**

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

**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

## **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**

Select sid, name from sys.syslogins

Select sid, name from sys.sysusers

Select \* from sys.sysmember

Xong rồi hiện 3 cái table mình nhìn từng cái thuộc tính của nó rồi dóng qua tim fđược tên họ nhóm của nhân viên đó.

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

## **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?**

## **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

## **Câu 11: more attribute**

**- 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

**-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.

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

## **Câu 12: Có mấy loại JOIN**

+ INNER JOIN

+ OUTER JOIN (FULL JOIN): Left Join & Right Join

+ LEFT JOIN,

+ RIGHT JOIN

# **CÂU HỎI NGẮN**

## **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

## **Dữ liệu rác là gì?**

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

## **Tại sao khi gõ login lại biết họ tên? / Tại sao biết user liên kết với login nào?**

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

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

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

## **Loginname nằm trong table nào?**

sys.sysuser

## **Tên nhóm quyền nằm trong table nào?**

Sys.sysuser

## **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.

## **Chuyển chữ thường thành chữ hoa**

dùng lệnh **upper**

## **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.

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

Đẩy về server publicsher sau đó đồng bộ xuống các Subcriber

## **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.

## **Subform là gi?**

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

## **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)

## **Dữ liệu sau khi nhập form sẽ được đẩy về đâu**

đẩy về publisher sau đó đồng bộ xuống các subscriber

## **Tại sao dùng remote login**

1. **Chỉ ra phân mảnh dọc trong đề tài?**

* Dọc: server 3 QLVT được đặt trên server 3: chứa thông tin các nhân viên, kho của cả 2 chi nhánh 1 và 2. Server này dùng để tra cứu thông tin của nhân viên, kho của cả 2 chi nhánh

## **Viết Stored Procedure ở đâu là hợp lý ?**

Đáp: Nên viết ở Server gốc. Sau đó sẽ tùy vào từng chức năng mà đẩy xuống các server phân mảnh là hợp lý nhất.

1. **Sau khi phân tán xong thì có 1 trường dữ liệu là rowguild. Vậy rowguid 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 một server.

Username là tài khoản mà chúng ta dùng để làm việc trên một cơ sở dữ liệu.

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

chúng là những quyền mà một tài khoản kiểu login name có thể được chỉ định & quyết định xem chúng có thể làm những gì trên cơ sở dữ liệu đó.

**1.db\_dataReader** là quyền chỉ xem dữ liệu, không thể thêm mới hoặc sửa đổi dữ liệu

**2.db\_dataWriter** là quyền chỉ ghi dữ liệu, dĩ nhiên nếu ghi được thì coi như cũng đọc được dữ liệu

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

**4.db\_accessAdmin** là quyền tạo các tài khoản username, tài khoản dùng để thao tác trên một cơ sơ dữ liệu nhất định

**5.db\_owner** là quyền cao nhất với một cơ sở dữ liệu, quyền này cho phép xem, thêm, xóa, sửa tất cả dữ liệu & có thể tạo tài khoản login name và username mà không có giới hạn nào( kiểu bạn là chủ nhà thì làm gì cũng được ấy )

1. **Có mấy cách để viết Stored Procedure?**

Có 2 cách: viết trên từng site và viết trên site chủ sau đó đồng bộ

C2 hay hơn do C1 muốn sửa thì phải sửa thủ công trên từng site

1. **Nếu sửa Stored Procedure 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 một 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 site khác

1. **Muốn thực thi một câu lệnh store procedure, view, function thì ta làm như nào ?**

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

1. **ý nghĩa của Begin trans, commit, rollback & 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. **Khi viết 1 stored procedure khi nào ta không dùng begin transaction, không dùng commit ... nhưng vẫn được coi là 1 giao tác?**

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

1. **Run on continue khác run on demand ở điểm nào?**

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

1. **Snapshot folder là gì?**

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 đẩy qua đẩy lại. Phải là network path(shared folder)

1. **Tại sao trả về mã nhân viên khi dùng sp\_DangNhap?**

để gán tự động cho các form có mã nhân viên

1. **Vì sao ta phải dùng Remote Login?**

Cho phép truy cập dữ liệu khi đứng từ server này sang server khác.

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

Đẩy về publisher sau đó đồng bộ xuống các subcriber.



## **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.