

1. cho biết sự khác nhau giữa mô hình ERD và CD

Các sơ đồ ER (mối quan hệ thực thể) và sơ đồ Lớp là hai trong số các sơ đồ thiết kế mà các nhà phát triển phần mềm tạo ra thường trong các giai đoạn thiết kế của vòng đời kỹ thuật phần mềm. Các sơ đồ ER là một sản phẩm của kỹ thuật mô hình hóa mối quan hệ thực thể (ERM) để mô hình hóa cơ sở dữ liệu. Một sơ đồ lớp, được viết bằng Ngôn ngữ mô hình hóa thống nhất, là một sơ đồ mô tả cấu trúc của một hệ thống được đề xuất. Mặc dù không có yêu cầu phải có một ánh xạ chính xác từ một đến một giữa các lớp trong sơ đồ lớp và các thực thể trong sơ đồ thực thể, nhưng nhìn chung có một số mối quan hệ có ý nghĩa giữa chúng. Tuy nhiên, có rất nhiều trường hợp một thực thể của sơ đồ ER ánh xạ tới nhiều lớp của sơ đồ lớp tương ứng hoặc một lớp duy nhất của sơ đồ lớp ánh xạ tới nhiều thực thể của sơ đồ ER tương ứng. Nhưng,

Sơ đồ ER là gì?

Sơ đồ ER là sản phẩm của mô hình hóa mối quan hệ thực thể. Mô hình hóa mối quan hệ thực thể là quá trình đưa ra một biểu diễn trừu tượng và khái niệm dữ liệu. Sơ đồ ER cuối cùng mô hình hóa các cơ sở dữ liệu. Cụ thể hơn, nó tạo ra lược đồ khái niệm của mô hình dữ liệu. Các khối xây dựng chính của sơ đồ ER là các thực thể, mối quan hệ và thuộc tính. Một thực thể đại diện cho một thứ có thể tồn tại độc lập và có thể được định nghĩa duy nhất. Thông thường, một thực thể đại diện cho một đối tượng trong thế giới thực như xe hơi hoặc nhân viên. Các thực thể có thể mặc dù là danh từ xuất hiện trong quá trình mô tả vấn đề cần giải quyết. Mối quan hệ cho thấy cách các thực thể được kết nối. Chúng giống như các động từ được tìm thấy trong phần mô tả vấn đề cần giải quyết. Thuộc tính của cả thực thể và thuộc tính được gọi là thuộc tính.

Sơ đồ lớp là gì?

Một sơ đồ lớp (được gọi chính xác hơn là sơ đồ lớp UML) là sơ đồ thiết kế đại diện cho cấu trúc tĩnh và hành vi của một hệ thống được đề xuất, được xác định bằng UML (Ngôn ngữ mô hình hóa thống nhất). Một sơ đồ lớp cho thấy các lớp của hệ thống, mối quan hệ giữa các lớp và các thuộc tính của chúng. Các lớp mô tả biểu diễn trừu tượng của các đối tượng trong thế giới thực, trong khi các mối quan hệ mô tả cách mỗi lớp được kết nối với các lớp khác. Cả hai lớp và các mối quan hệ có các thuộc tính được gọi là thuộc tính. Các phương thức trong các lớp đại diện hoặc định nghĩa hành vi của các lớp này. Các phương thức và thuộc tính của các lớp được gọi là các thành viên của lớp.

Sự khác biệt giữa Biểu đồ ER và Biểu đồ lớp là gì?

Mặc dù sơ đồ ER và sơ đồ lớp là hai trong số các sơ đồ thiết kế mà các nhà phát triển thường gặp trong các giai đoạn thiết kế của các dự án kỹ thuật phần mềm, chúng có những điểm khác biệt chính. Các sơ đồ ER đại diện cho biểu diễn trừu tượng của mô hình dữ liệu, trong khi các sơ đồ lớp biểu thị cấu trúc tĩnh và hành vi của hệ thống được đề xuất. Các khối xây dựng chính của sơ đồ ER là các thực thể, mối quan hệ và thuộc tính nhưng các khối xây dựng chính của sơ đồ lớp là các lớp, mối quan hệ và thuộc tính. Biểu đồ lớp có nhiều khả năng ánh xạ vào các đối tượng trong thế giới thực, trong khi sơ đồ ER thường ánh xạ vào các bảng trong cơ sở dữ liệu. Thông thường, các mối quan hệ được tìm thấy trong sơ đồ ER khó hiểu đối với con người hơn các mối quan hệ trong sơ đồ lớp.

2. vì sao cần back up dữ liệu, có bao nhiêu cách backup

Backup dữ liệu là sao chép dữ liệu máy chủ (hoặc máy tính cá nhân, điện thoại, máy tính bảng,...) rồi lưu trữ ở một nơi khác, phòng khi có sự cố xảy ra như thiên tai, virus, ổ cứng hỏng,...sẽ không bị mất dữ liệu.

Các Dạng Backup

Full Backup là gì?

Full Backup có nghĩa là backup toàn bộ dữ liệu đang có của bạn

Ưu điểm của Full Backup :

- Dễ dàng phục hồi lại dữ liệu. Khi cần phục hồi lại thì sẽ phục hồi lại toàn bộ dữ liệu của ngày **Backup Full**.
- Tính an toàn cao cho dữ liệu.

Nhược điểm của Full Backup:

- Thời gian backup lâu. Dữ liệu càng nhiều thì thời gian backup càng lâu
- Tốn dung lượng lưu trữ. Nếu mỗi ngày đều chạy full backup thì rất tốn dung lượng ổ cứng. Ví dụ một ổ cứng 1TB nếu backup full mỗi ngày thì trong 1 tuần sẽ cần 7TB lưu trữ.
- Chi phí đầu tư thiết bị lưu trữ lớn.

Differential Backup là gì ?

Differential Backup là backup những gì thay đổi so với lần **Full Backup** gần nhất

Ưu điểm của Differential Backup :

- Thời gian backup nhanh hơn
- Dung lượng backup nhỏ hơn so với Full Backup. Tiết kiệm dung lượng lưu trữ
- Tốc độ phục hồi dữ liệu sẽ nhanh hơn so với **Incremental Backup**

Nhược điểm của Differential Backup :

- Khi cần khôi phục dữ liệu cần có 2 bản backup :
 - 1 File **Full Backup** lần gần nhất và 1 File **Differential Backup** vào thời điểm cần restore

Incremental Backup là gì :

Incremental Backup là backup những gì thay đổi so với lần **Incremental Backup** gần nhất

Ưu điểm của Incremental Backup :

- Thời gian backup nhanh nhất
- Dung lượng backup bé nhất

Nhược điểm của Incremental Backup :

- Khi cần khôi phục dữ liệu phải có đủ các bản backup :
 - 1 File Full backup lần gần nhất và **tất cả** các File **Incremental Backup** từ thời điểm **Full Backup** đến thời điểm cần restore
- Thời gian Restore lâu hơn so với Differential Backup

Giải pháp được các nhà backup khuyến cáo nên lựa chọn thường như sau :

Full Backup hàng tháng.

Differential Backup hàng tuần.

Incremental Backup hàng ngày.

3. để đảm bảo tính an ninh cho csdl, cần dùng backup gì

Câu này chưa có câu trả lời

- Chu kỳ n/min (n lần/ phút) – partial backup
- Cuối ngày – full backup

4. cho biết sự khác nhau csdl tập chung và phân tán

CSDL tập trung	CSDL phân tán
<p>– Toàn bộ CSDL được lưu ở một dàn máy tính (có thể bao gồm một hoặc nhiều máy được điều hành chung như một máy).</p> <p>– Đặc trưng:</p> <ul style="list-style-type: none"> + Dữ liệu được lưu trên máy tính trung tâm + Nhiều người dùng từ xa có thể truy cập CSDL. <p>– Phân loại:</p> <ul style="list-style-type: none"> + Hệ CSDL cá nhân + Hệ CSDL trung tâm + Hệ CSDL khách chủ 	<p>– Dữ liệu không lưu trữ tập trung ở một máy mà được lưu trữ ở nhiều máy trên mạng và được tổ chức thành những CSDL con.</p> <p>– Đặc trưng:</p> <ul style="list-style-type: none"> + Dữ liệu có thể được lưu trữ ở nhiều CSDL con + Có thể sử dụng dữ liệu ở nhiều nơi để truy vấn. <p>– Phân loại:</p> <ul style="list-style-type: none"> + Hệ CSDL thuần nhất + Hệ CSDL hỗn hợp

5. cho biết sự khác phân mảnh ngang phân mảnh dọc cho ví dụ

Phân mảnh ngang: qh đc chia theo chiều ngang, thực chất đây là phép chọn trong quan hệ. chọn những bộ của qh thỏa mã một biểu thức đk cho trước

Phân mảnh ngang chính là việc chia quan hệ thành nhiều các nhóm bộ. Kết quả của quá trình phân mảnh ngang là các quan hệ con, số lượng quan hệ con phụ thuộc vào điều kiện ràng buộc của các thuộc tính. Và các bộ trong các quan hệ con là tách biệt nhau. Phân mảnh ngang thực chất là phép chọn quan hệ thỏa mãn một biểu thức điều kiện cho trước.

Có hai loại phương pháp phân mảnh ngang là:

- Phân mảnh ngang nguyên thủy: Là phân mảnh ngang được thực hiện trên các vị từ của chính quan hệ đó.
- Phân mảnh ngang dẫn xuất: Là phân rã một quan hệ dựa trên các vị từ của quan hệ khác.

Phân mảnh dọc: các quan hệ được chia theo chiều dọc. nghĩa là thiết lập một qh mới chỉ có 1 số thuộc tính từ quan hệ gốc, thực chất đây là phép chiếu trên tập con các thuộc tính của quan hệ

2.4.1 Khái niệm phân mảnh dọc

Phân mảnh dọc quan hệ R sinh ra các mảnh R_1, R_2, \dots, R_r , sao cho mỗi mảnh chứa một tập con các thuộc tính của quan hệ R và khoá của nó. Mục đích của phân mảnh dọc là phân chia quan hệ R thành tập các quan hệ nhỏ hơn để có nhiều ứng dụng có thể chỉ cần thực hiện trên một mảnh. Mảnh tối ưu là mảnh sinh ra một lược đồ phân mảnh cho phép giảm tối thiểu thời gian thực hiện của ứng dụng trên mảnh đó.

6. sự khác nhau giữa csdl thông thường và di động. cho ví dụ về csdl di động, trên thiết bị di động, cấu trúc 1 vài bảng về csdl đó

CSDL di động

- ☐ Một cơ sở dữ liệu di động là một cơ sở dữ liệu có thể được kết nối với một thiết bị máy tính bằng điện thoại di động thông qua mạng điện thoại di động không dây.
- ☐ Đặc điểm CSDL di động:
 - tách biệt về vật lý từ các máy chủ cơ sở dữ liệu trung tâm.
 - cư trú trên các thiết bị di động.
 - Có khả năng giao tiếp với một máy chủ cơ sở dữ liệu trung ương hoặc các máy di động khác từ các site từ xa.
 - Xử lý các truy vấn cục bộ không có kết nối.
- ☐ **Mô hình client-server Mobile Databases**
 - Mô hình client-server là mô hình truyền thống của các hệ thống thông tin.
 - Đây là mô hình chủ đạo cho cơ sở dữ liệu điện thoại di động hiện có.
 - Các máy chủ là một điểm duy nhất nên có thể tạo ra hiện tượng nút cổ chai.
- ☐ **Mô hình Peer-to-Peer Mobile Databases**
 - Trong cơ sở dữ liệu điện thoại di động P2P, các hoạt động đảm bảo cơ sở dữ liệu được phân bổ cho các khách hàng.

- Mỗi quá trình đóng một phần vai trò của máy chủ, bên cạnh vai trò của khách hàng.
- Một khách hàng nào muốn truy cập vào một mảnh của dữ liệu, gửi một yêu cầu cho khách hàng khác và họ chuyển tiếp yêu cầu đến khi dữ liệu được tìm thấy.
- Các vấn đề lớn trong mô hình này là đảm bảo sự sẵn có của dữ liệu

Đặc điểm của môi trường di động

- ☐ Băng thông hạn chế của các mạng không dây.
- ☐ Nguồn điện hạn chế.
- ☐ Nguồn lực hạn chế.
- ☐ Mobility.
- ☐ Ngắt kết nối (disconnect)
- ☐ Bất cứ nơi nào: có thể di chuyển đối tượng mà không ảnh hưởng đến dữ liệu sẵn có
- ☐ Bất cứ lúc nào: có thể lưu trữ và truy xuất dữ liệu bất cứ khi nào kết nối có sẵn
- ☐ Hiệu quả:
 - ☐ Có thể xử lý dữ liệu điện thoại di động hiệu quả
 - ☐ Có thể cung cấp cho người dùng những gì họ thực sự muốn (và bỏ qua ít có liên quan ...)
- ☐ Băng thông không dây và tốc độ truyền thông hạn chế
- ☐ Kích thước màn hình làm giới hạn output
- ☐ Khả năng truy vấn hạn chế đối với đầu vào
- ☐ Hạn chế tính toán & dung lượng lưu trữ
- ☐ An ninh yếu

Cách tiếp cận

- ☐ Hiện nay hầu hết các nhà phát triển ứng dụng di động sử dụng "các tập tin phẳng" để lưu trữ dữ liệu ứng dụng.
- ☐ Một "tập tin phẳng" là một tập tin có chứa các bản ghi mà không có mối tương quan có cấu trúc.
- ☐ Ưu điểm: nhỏ và dễ dàng hơn để quản lý.
- ☐ Nhược điểm: Các ứng dụng cần phải biết cách tổ chức của các bản ghi trong tập tin. Các nhà phát triển phải thực hiện các chức năng của một cơ sở dữ liệu cần thiết.

Các yêu cầu của một MDB

- ☐ DBMS Mobile phải thỏa mãn các yêu cầu sau:
 - Bộ nhớ nhỏ.
 - Tối ưu hóa hệ thống lưu trữ.
 - Đồng bộ dữ liệu.
 - An ninh.
 - Tiêu thụ điện năng thấp.
 - Tự quản lý.
 - Nhúng trong ứng dụng
- ☐ Có những tính năng:
 - Kích thước nhỏ.
 - Đề kháng tốt với sốc.
 - Tiêu thụ điện năng thấp.
 - Thời gian truy cập nhanh.

Đồng bộ hóa dữ liệu

- ☐ Thiết bị cầm tay không thể kết nối tất cả các thời gian.

- ☐ Người dùng có thể truy cập và thao tác dữ liệu trên các thiết bị của họ.
- ☐ Họ cũng không thể lưu trữ một lượng lớn dữ liệu do thiếu khả năng lưu trữ.
- ☐ DBMS di động nên có chức năng đồng bộ hóa để tích hợp các phiên bản khác nhau của dữ liệu vào một phiên bản phù hợp.

An toàn (security)

- ☐ Dữ liệu được truyền qua một mạng không dây cần đặt ra các vấn đề an ninh.
- ☐ DBMS Mobile nên thực hiện một sự an toàn tuyệt end-to-end để đảm bảo việc chuyển giao an toàn của dữ liệu.

Tiêu thụ điện năng thấp

- ☐ Tuổi thọ pin của điện thoại di động dự kiến sẽ chỉ tăng 20% trong 10 năm tiếp theo.
- ☐ Bộ vi xử lý, hiển thị và kết nối mạng là những tiêu hao điện năng chính trong một thiết bị di động.
- ☐ DBMS di động cần phải được tối ưu hóa cho tiêu thụ năng lượng hiệu quả.

Tự quản lý (self-management)

- ☐ Trong cơ sở dữ liệu truyền thống, người quản trị cơ sở dữ liệu (DBA) là chịu trách nhiệm cho bảo trì cơ sở dữ liệu.
- ☐ Trong DBMS di động có thể không có DBA để quản lý cơ sở dữ liệu.
- ☐ DBMS di động cần phải hỗ trợ tự quản lý và tự động thực hiện các nhiệm vụ DBA.
- ☐ Một số DBMS di động cho phép quản lý từ xa.

Có thể nhúng trong các ứng dụng

- ☐ Các quản trị viên không có quyền truy cập trực tiếp đến các thiết bị di động.
- ☐ DBMS Mobile là một phần không thể thiếu của các ứng dụng, nó có thể được cung cấp như một phần của ứng dụng.
- ☐ Các cơ sở dữ liệu phải được nhúng như là một tập tin DLL trong ứng dụng.
- ☐ Các khả năng cần thiết của DBMS di động bao gồm khả năng:
 - giao tiếp với máy chủ cơ sở dữ liệu tập trung thông qua các chế độ như truy cập không dây hoặc Internet;
 - sao chép dữ liệu trên máy chủ cơ sở dữ liệu tập trung và thiết bị di động;
 - đồng bộ hóa dữ liệu trên máy chủ cơ sở dữ liệu tập trung và thiết bị di động;
 - thu thập dữ liệu từ các nguồn khác nhau như Internet;
 - quản lý/phân tích dữ liệu trên thiết bị di động;
 - tạo ra các ứng dụng di động tùy chỉnh

Mobile databases

- ☐ Sybase SQL Anywhere
- ☐ Oracle Lite
- ☐ Microsoft SQL Server Compact
- ☐ SQLite
- ☐ IBM DB2 Everyplace (DB2e)

Sybase SQL Anywhere

- ☐ SQL Anywhere đã được đưa ra trong năm 1995.
- ☐ Nó thống trị lĩnh vực cơ sở dữ liệu DD với khoảng 68% thị trường.
- ☐ File cơ sở dữ liệu độc lập với hệ điều hành và chuyển giao giữa các HĐH được hỗ trợ.
- ☐ Mã hóa được hỗ trợ mạnh mẽ

Oracle Lite

- ☐ Omniscience Object Technology, Inc đã được mua lại bởi tập đoàn Oracle 1996.
- ☐ Oracle Lite chạy ở dưới 1 MB bộ nhớ, và có thể được cài đặt trong 3 MB không gian đĩa cứng.
- ☐ Oracle Lite dùng cơ sở dữ liệu quan hệ một người dùng.

Microsoft SQL Server Compact

- ☐ Trước đây được gọi là SQL Server Mobile Edition.
- ☐ SQL Server Compact là miễn phí để tải về và phân phối.
- ☐ Nó cần một bộ nhớ trong khoảng 2MB, 5 MB cho lưu trữ và thực thi

SQLite

- ☐ SQLite là một cơ sở dữ liệu điện thoại di động mã nguồn mở.
- ☐ Nó là một cơ sở dữ liệu mà ít cần cấu hình với máy chủ.
- ☐ SQLite là một lựa chọn phổ biến như cơ sở dữ liệu di động.
- ☐ SQLite thực hiện hầu hết các tiêu chuẩn SQL-92.

DB2e

- ☐ Nó có thị phần lớn nhất sau khi SQL Anywhere.
- ☐ Nó có bộ nhớ nhỏ nhất (350 KB) trong so sánh với cơ sở dữ liệu điện thoại di động thương mại khác.
- ☐ DB2e đã bị ngưng vào tháng 4, 2013.
- ☐ IBM đã thay thế DB2e với IBM solidDB.
- ☐ SolidDB là một trong MDBMS có các tính năng trên dữ liệu khá tốt.

CSDL nhúng

- ☐ Họ được tối ưu hóa cho các thiết bị cụ thể như thẻ thông minh và cảm biến.
- ☐ Hệ thống cơ sở dữ liệu nhúng được tích hợp chặt chẽ với các ứng dụng.
- ☐ Chúng được ẩn từ đối với người dùng cuối.
- ☐ Cơ sở dữ liệu nhúng cần nguồn lực ít hơn khi so sánh với cơ sở dữ liệu di động.
- ☐ Chúng hỗ trợ các chức năng khá hạn chế và tuân các quy định của SQL chuẩn.
- ☐ PicoDBMS hỗ trợ đầy đủ các chức năng cho ứng dụng trên thẻ thông minh.
- ☐ Ứng dụng thẻ thông minh được sử dụng cho việc quản lý dữ liệu như chèn, xóa, cập nhật và tìm kiếm.
- ☐ PicoDBMS hỗ trợ một phần của SQL:
 - ☐ INSERT, UPDATE, DELETE, CHỌN
 - ☐ CREATE / DROP TABLE / XEM
 - ☐ GRANT / thu hồi
- ☐ Kích thước của PicoDBMS: 30KBytes.
- ☐ TinyDB đã được phát triển tại Đại học Berkeley.
- ☐ Nó chỉ hỗ trợ các chức năng cần thiết cho các ứng dụng cảm biến.
- ☐ Hầu hết các ứng dụng cảm biến được sử dụng để lọc ra một số dữ liệu với điều kiện nhất định.
- ☐ TinyDB chỉ hỗ trợ hoạt động SELECT của SQL chuẩn.

□ Bộ nhớ của nó: 3Kbytes

	TinyDB	PicoDBMS	Oracle Lite	IBM DB2e	MS SQL Server Compact
Small Footprint	Y	Y	Y	Y	Y
Flash-Optimized Storage System	N	N	N	N	N
Data Synchronization	N	N	Y	Y	Y
Self-Management	Y	Y	N/A	Y	N/A
Low Power Consumption	Y	Y	N	Y	Y
Security	N	Y	Y	Y	Y

	TinyDB	PicoDBMS	Oracle Lite	IBM DB2e	MS SQL Server Compact
Minimum Footprint Size	3 KB	30 KB	970 KB	320 KB	2 MB
SQL	SELECT only	a part of SQL99	a part of SQL99	a part of SQL99	a part of SQL99
Views	N	Y	Y	Y	Y
Integrity Constraints	N	N/A	Y	Y	Y
Concurrency	N	N	Y	Y	Y
Indexing	N	Y	Y	Y	Y
Encryption	N	N/A	Y	Y	Y
Access Control	N	Y	Y	Y	Y

Các vấn đề mà MDB cần xử lý

- Xử lý truy vấn
- Điều khiển tính đồng thời
- Quản lý giao tác và khôi phục

Xử lý truy vấn

- ☐ Các loại Query
 - Location dependent query
 - Location independent query
- ☐ Các ràng buộc
 - Query Query response time
 - Search-result accuracy
 - Throughput: number of queries per time unit

Các thách thức ở tương lai

- ☐ Chỉ mục và lưu trữ (Indexing and storage of mobile data)
- ☐ Rút trích và xử lý truy vấn hiệu quả (Efficient query processing and retrieval of mobile data)
- ☐ Hiển thị và thích nghi thông tin với kích thước hạn chế (Content adaptation and information presentation for small display)
- ☐ Thông kê và cá nhân hóa dữ liệu (Automatic summarization and personalization of mobile data)
- ☐ Scalable (mở rộng) algorithms for large mobile databases
- ☐ Chia sẻ tri thức giữa các thiết bị di động (Knowledge sharing among mobile peers)
- ☐ An toàn dữ liệu trên TBDD (Security and trusted retrieval of mobile data)
- ☐ Tìm kiếm trên nền tảng vị trí (Location-based search for mobile devices)

7. cho biết sự khác nhau giữa nosql và sql, nosql có bao nhiêu loại và liệt kê, đồ thị, văn bản, khóa giá trị., liệu kê và mô tả

So sánh đang còn ở trong trang a4 đang còn nhớ chép thêm

Bảng 1. So sánh sự giống nhau và khác nhau giữa SQL và NoSQL

	SQL	NoSQL
Loại CSDL	Mỗi loại CSDL SQL có các biến thể nhỏ	Nhiều loại CSDL NoSQL khác nhau, bao gồm: CSDL cặp khóa – giá trị, CSDL hướng tài liệu, CSDL hướng cột và CSDL hướng đồ thị
Mục đích phát triển	Đáp ứng nhu cầu lưu trữ đầu tiên của các ứng dụng	Đáp ứng nhu cầu vượt qua ngoài khả năng đáp ứng của SQL, đặc biệt quan tâm đến quy mô của dữ liệu, phát triển và lưu trữ dữ liệu phi cấu trúc.
Các hệ CSDL điển hình	Microsoft SQL Server, MySQL, Postgre, Oracle Database	MongoDB, Cassandra, Hbase, Neo4j
Mô hình dữ liệu	Cấu trúc lưu trữ xây dựng trước khi lưu trữ dữ liệu	Không cần xây dựng sẵn cấu trúc lưu trữ, việc cần làm là tiến hành lưu trữ dữ liệu vào
Khả năng mở rộng	Chỉ có thể mở rộng theo chiều dọc	Có thể mở rộng theo chiều dọc, chiều ngang đồng thời hỗ trợ công nghệ điện toán đám mây
Mã nguồn	Cả đóng và mở	Chỉ có mã nguồn mở
Hỗ trợ	Hỗ trợ tốt cho khách hàng	Hỗ trợ trong một số trường hợp với một

Những loại của nosql

- Tính đến thời điểm này có đến 150 loại CSDL NoSQL, chia thành 13 mô hình lưu trữ dữ liệu khác nhau.
- Có 4 loại CSDL NoSQL thông dụng như:
 - CSDL hướng cột,
 - CSDL cặp khoá – giá trị,
 - CSDL đồ thị,
 - CSDL hướng tài liệu [1].

8. mô hình dữ liệu 2 mức theo mô hình hướng đối tượng.

9. csdl đa phương tiện, phân tán nosql, di động,(csdl di động đã có ở câu 6)

9.1 CSDL Đa phương tiện(MMDB)

Đoạn đầu này gt cái mẹ gì đó cx đéo biết, kéo xuống dưới xem 9.1.1

Một số khái niệm

- Media (tiếng Latin: *medius*, tiếng Anh: *means, intermediary*) là đề cập đến các loại dữ liệu, thông tin ở dạng: văn bản, ảnh, âm thanh và video.
- *Phân loại media*: Có nhiều cách phân loại, nhưng cách chung nhất là phân loại trên cơ sở các quan hệ media với thời gian. Qui định này dẫn tới hai lớp media: tĩnh (*static*) và động (*dynamic*).
 - *Static media*: Không có chiều thời gian, nội dung và ý nghĩa của chúng không phụ thuộc vào thời gian trình diễn. Media tĩnh bao gồm dữ liệu văn bản, đồ họa.
 - *Dynamic media*: Có chiều thời gian, ý nghĩa và độ chính xác của chúng phụ thuộc vào tốc độ trình diễn. *Dynamic media* bao gồm *animation, video, audio*.
- **Multimedia**
 - Khái niệm *multimedia* (tiếng Latin: *multus*- tiếng Anh: *numerous*) đề cập đến tập hợp các kiểu media được sử dụng chung, trong đó ít nhất có một kiểu media không phải là văn bản (nói cách khác là ít nhất có một media trong đó là ảnh, audio hay video).
- Khái niệm *multimedia có thể đi kèm với cụm từ khác nhau*: thông tin đa phương tiện, dữ liệu đa phương tiện, hệ thống đa phương tiện, truyền thông đa phương tiện, ứng dụng đa phương tiện
- Khái niệm dữ liệu đa phương tiện đề cập đến sự biểu diễn các kiểu media khác nhau mà máy tính có thể đọc được. Thông tin đa phương tiện đề cập đến thông tin được truyền đạt bởi các kiểu media.

Mô tả sơ lược dữ liệu MM và các tính chất của chúng

- Chúng ta đang đối mặt với sự bùng nổ thông tin đa phương tiện.
 - Một số lượng lớn ảnh và video trên Internet.
 - Rất nhiều tranh vẽ, ảnh chụp đang được chuyển sang dạng số để dễ xử lý, bảo quản.
 - Các bức ảnh từ bản tin truy vấn và trên báo cũng đang được chuyển sang dạng số để dễ dàng quản lý.
 - Lượng lớn ảnh y tế, ảnh vệ tinh đang được thu thập hàng ngày.
 - Xu thế này đã thúc đẩy phát triển công nghệ số lưu trữ và trình diễn.
 - Không thể sử dụng nhanh và hiệu quả các thông tin đa phương tiện này nếu chúng không được tổ chức tốt để có khả năng truy tìm nhanh.
 - Không chỉ khối lượng dữ liệu đa phương tiện lưu trữ tăng nhanh mà các kiểu dữ liệu và đặc tính của chúng khác xa dữ liệu văn bản và số.

Đặc trưng

- ☐ Khối lượng khổng lồ (đặc biệt với dữ liệu audio và video).
- ☐ Audio và video có thêm chiều thời gian.
- ☐ Dữ liệu ảnh, audio và video được thể hiện bởi dãy các giá trị mẫu, không có cấu trúc nhất định để máy tính tự động nhận biết.
- ☐ Rất nhiều ứng dụng đa phương tiện đòi hỏi trình diễn đồng thời các loại media khác nhau. Thí dụ, phim bao gồm các ảnh đồng bộ với âm thanh.
- ☐ Ý nghĩa của dữ liệu đa phương tiện đôi khi rất mờ.
- ☐ Dữ liệu đa phương tiện rất giàu thông tin. Đòi hỏi nhiều tham số để biểu diễn nội dung của chúng.
- ☐ Một MDBMS cần phải đảm bảo các tính năng cơ bản của một DBMS, ngoài ra nó còn phải có các tính năng khác như:
 - ☐ Khả năng khai thác dữ liệu..
 - ☐ Khả năng lưu trữ lớn.
 - ☐ Giao diện multimedia.
 - ☐ Hỗ trợ truy vấn tương tác multimedia.

9.1.1. CSDL đa phương tiện

- ☐ DBMS thông thường không đáp ứng đầy đủ
 - Yêu cầu truy tìm
 - Chỉ số hóa đa phương tiện
- ☐ Do vậy, đòi hỏi kỹ thuật mới để quản lý các tính chất đặc biệt của dữ liệu đa phương tiện.
- ☐ Tuy nhiên ta nhận ra rằng DBMS và IR có thể đóng vai trò quan trọng trong MMDBMS.
- ☐ Nhiều phần dữ liệu đa phương tiện như ngày tạo lập, tác giả, v.v.. là có cấu trúc.
- ☐ Chúng có thể được quản lý bằng các kỹ thuật DBMS.
- ☐ Mô tả (*annotation*) bằng văn bản vẫn còn là phương pháp hiệu quả để thu thập nội dung dữ liệu đa phương tiện, do vậy các kỹ thuật IR vẫn đóng vai trò quan trọng.
- ☐ Tóm lại, cần phải tích hợp DBMS, IR và các kỹ thuật đặc biệt khác quản lý dữ liệu đa phương tiện để phát triển MIRS phù hợp và hiệu quả.
- ☐ Để hỗ trợ cho các trường có giá trị lớn trong RDBMS, một khái niệm được gọi là đối tượng rộng hoặc nhị phân (BLOB) được giới thiệu.
- ☐ Một BLOB là một chuỗi bit lớn các độ dài biến.
 - Ví dụ, nếu ta muốn lưu bức tranh của sinh viên trong bản ghi ở bảng STUDENT trên, chúng ta có thể tạo ra một bảng :
- ☐ *Create table STUDENT(*
 - *IDStu# integer,*
 - *Name char(20),*
 - *address char(100),*
 - *Picture BLOB);*
- ☐ Các BLOB bình thường chỉ là chuỗi bit và hoạt động bằng việc so sánh.
- ☐ RDBMS không biết nội dung hoặc ngữ nghĩa của một BLOB.
- ☐ Tất cả BLOB hiểu như một khối dữ liệu.
- ☐ Điểm khác biệt chính giữa BLOB và các đối tượng:
 - đối tượng thì được định rõ một cách thích đáng, bao gồm các đặc tính và cho phép chúng có tác dụng
 - BLOB thì không.
- ☐ Các khái niệm về các BLOB và các đối tượng là một bước gần với xử lý dữ liệu multimedia .
- ☐ Nhưng các BLOB được sử dụng chỉ để lưu dữ liệu có khối lượng lớn.
- ☐ Trong khi các đối tượng chứa vài thuộc tính đơn giản, nhiều chức năng hơn nên được phát triển để xử lý việc truy xuất multimedia dựa vào nội dung. Một vài các khả năng được yêu cầu như sau:
- ☐ Các công cụ, tự động hoá hoặc bán tự động trích rút các nội dung và các đặc trưng trong dữ liệu multimedia ;
- ☐ Các cấu trúc chỉ số hoá, để điều khiển các vector multimedia

- ☐ Các độ đo tương đồng, nhằm truy xuất multimedia thay vì kết nối một cách chính xác
- ☐ Lưu trữ các hệ thống phụ, thiết kế lại nhằm đáp ứng các yêu cầu của băng tần cao với cỡ lớn, thoả mãn những đòi hỏi theo kiểu thời gian thực
- ☐ Giao diện sử dụng, được thiết kế cho phép các câu hỏi đa dạng trong nhiều kiểu truyền thông đa dạng và cung cấp các trình bd multimedia.
- ☐ Thuộc. Một bác sĩ đưa ra ảnh chụp sóng siêu âm mới và muốn tìm lại ảnh với mức độ có thể so sánh được của sự phình tâm thất trái từ một ảnh chụp siêu âm cơ sở
- ☐ Bảo mật. Một cảnh sát đưa ra hệ thống với một bức tranh vẽ mặt người và muốn phục hồi lại những hình ảnh khác vào hồ sơ hiện thời của những người giống với bức tranh này từ cơ sở thông tin bảo mật
- ☐ Giáo dục. Một sinh viên quét chụp một tranh động vật và muốn truy tìm tất cả các thông tin (bao gồm âm thanh, hình ảnh, văn bản mô tả). Tiếp theo, một sinh viên khác thêm âm thanh cho con vật và muốn khôi phục lại bức tranh và thông tin mô tả loại động vật này

Các MIRS và nhiều loại câu hỏi khác nhau

- ☐ Báo chí. Một báo cáo viên viết một bài báo về một người và muốn truy tìm bức tranh của người đó và các thông tin tổng hợp đã xuất hiện trên các báo và tivi 20 năm trước
- ☐ Giải trí. Một quan sát viên muốn truy tìm một đoạn băng tương tự với những gì mà anh ta đã xem trên một cơ sở dữ liệu video lớn hơn
- ☐ Đăng ký nhãn hiệu. Một cán bộ gia công một nhãn hiệu đăng ký nào đó muốn quyết định xem đã có một cái nhãn hiệu nào trước đó giống như vậy đã được đăng ký chưa. Để làm điều này, anh ta cần một cơ sở dữ liệu về nhãn hiệu để so sánh các nhãn hiệu giống nhau nhất hiện có để làm ra một nhãn hiệu hoàn toàn mới

Kiến trúc tổ chức nội dung

- ☐ Bao gồm 3 kiến trúc với 3 nguyên lý

Nguyên lý tự trị

- ☐ Nhóm tất cả các dữ liệu ảnh, dữ liệu video và tất cả các dữ liệu văn bản và chỉ số hóa chúng theo nguyên tắc tối đa hóa hiệu suất của tất cả các loại truy nhập đối với các loại dữ liệu mà chúng ta dự định.
- ☐ Nguyên lý này đảm bảo rằng với mỗi loại dữ liệu (ảnh, video, văn bản) chúng đều được tổ chức với một cách thức đặc trưng phù hợp với mỗi loại dữ liệu này.

Nguyên lý đồng nhất

- ☐ Nguyên lý này giúp tìm được một cấu trúc tóm tắt chung cho tất cả các loại dữ liệu.
- ☐ Cấu trúc này có thể được dùng trong việc chỉ số hóa tất cả các loại dữ liệu qua đó tạo ra một "chỉ số thống nhất" mà có thể dùng để truy cập tới các đối tượng khác nhau.
- ☐ Nói một cách khác là có thể trình bày tất cả các đối tượng khác nhau (ảnh, video, âm thanh, văn bản) trong một cấu trúc dữ liệu duy nhất và qua đó phát triển các thuật toán để truy vấn cấu trúc dữ liệu này.

Nguyên lý lai ghép

- ☐ Ý tưởng của nguyên lý này là dựa trên sự kết hợp của 2 nguyên lý đã trình bày.
- ☐ Kết quả của nguyên lý này là một kiểu dữ liệu nào đó sử dụng chỉ số (index) riêng của chúng, trong khi đó các kiểu dữ liệu khác sẽ sử dụng một chỉ số (Index) "thống nhất".

- Loại dữ liệu nào sử dụng kiểu chỉ số nào sẽ phụ thuộc vào các đặc tính khác nhau sẽ được nói đến ở phần sau.

Mô tả CSDL DPT cho tập ảnh, Video

Dữ liệu ảnh như là một tóm tắt media.

- Giả sử chúng ta xem một CSDL hình ảnh đơn giản gồm có 7 bức ảnh, CSDL này sẽ tương ứng với media trừu tượng sau:
 - Tập các trạng thái bao gồm (pic1.gif,.....,pic7.gif).
 - Tập các đặc trưng bao gồm tên của các đối tượng xuất hiện trong các bức ảnh tương ứng là Bob, Jim, Bill, Charlic và Ed.
 - Ảnh xạ thu gọn λ cho chúng ta biết với mỗi trạng thái có những đặc trưng nào xuất hiện trong nó. Bảng sau sẽ mô tả cho vấn đề này

Dữ liệu video như là một tóm tắt media.

- Tập các trạng thái bao gồm 5 frames video từ frame 1 đến frame 5.
 - Tập các đặc trưng bao gồm Jane Shady, Denis Dopeman, ngôi nhà của Dopeman và chiếc valy.
 - Ảnh xạ rút gọn λ được mô tả ở bảng sau
- Chúng ta có thể có một số quan hệ phụ thuộc vào trạng thái sau:
 - Quan hệ có (have) là một quan hệ phụ thuộc vào trạng thái xác định chiếc valy (đối tượng) xuất hiện cùng với nhân vật nào ở trong trạng thái nào? Quan hệ này có được mô tả như sau:

Một số hàm

Các hàm cơ bản của SMDSS (simple multimedia database systems) bao gồm:

- FindType(Obj): hàm này sử dụng đối tượng media Obj làm đầu vào và sẽ trả về kiểu của đối tượng, ví dụ:
 - FindType(image1.gif) = *gif*.
 - FindType(movie1.mpg) = *mpg*.
- FindObjWithFeature(f): hàm này sử dụng đặc trưng f là giá trị đầu vào và trả về một tập tất cả các loại đối tượng media mà có chứa đặc trưng f, ví dụ
- FindObjWithFeatureAttr(f,a,v): hàm này sử dụng đặc trưng f, một thuộc tính tên a kết hợp với đặc trưng này và một giá trị v là đầu vào. Giá trị trả về sẽ là tất cả các đối tượng o có chứa đặc trưng và giá trị của thuộc tính a trong đối tượng o là v. Ví dụ:
- FindFeaturesInObj(Obj): câu hỏi này đòi hỏi phải tìm tất cả các đặc trưng có mặt trong một đối tượng media đã cho. Kết quả trả về là một tập bao gồm tất cả các đặc trưng. Ví dụ:
- SMDSS-SQL mở rộng của chúng ta sẽ bao gồm:
 - Tất cả các câu lệnh chuẩn của SQL có
 - Bổ sung thêm một số đặc trưng của dữ liệu Multimedia vào SQL

Các hệ QTCSDL DPT

- Multimedia database management (NSF, Fuji Electric, AT&T)
 - Video modeling and management
 - Multimedia document management

- ☐ Distributed multimedia systems
(NSF, AFRL, IBM, Intel, Siemens)
- ☐ High-performance multimedia database architecture for storage management
(NSF, AT&T)

9.2 csdl (nosql) – phân tán trang cuối

- ☐ NoSQL là một thể hệ CSDL mới với các đặc điểm:
- ☐ Không ràng buộc, phân tán, mã nguồn mở, có khả năng mở rộng theo chiều ngang, lược đồ tự do.
- ☐ NoSQL có thể lưu trữ xử lý dữ liệu từ một lượng rất nhỏ cho đến hàng petabytes, trong một hệ thống chịu tải, chịu lỗi cao và đáp ứng thời gian thực

- ☐ Base và ACID
- ☐ ACID là viết tắt của cụm từ
 - Atomicity (nguyên tử),
 - Consistency (nhất quán),
 - Isolation (Cô lập), và
 - Durability (Lâu bền).
- ☐ Trong cơ sở dữ liệu NoSQL, các nguyên tắc của các mô hình ACID là quá mức cần thiết, trên thực tế nó đã cản trở hoạt động của các cơ sở dữ liệu.
- ☐ CSDL với các đặt tính của ACID là vô cùng khó khăn. Đó là lý do mà Consistency và Isolation bị thu hồi].
- ☐ **NoSQL dựa vào một mô hình nhẹ nhàng hơn, thích hợp hơn, và kết quả là chúng ta có phương pháp tiếp cận mới BASE gồm ba nguyên tắc:**
 - **Tính sẵn có cơ bản (Basic Availability)**
 - **Trạng thái mềm (Soft State)**
 - **Tính nhất quán cuối (Eventual Consistency).**
- ☐ Tóm lại các thuộc tính Base như sau:
 - một ứng dụng làm việc cơ bản tất cả thời gian (basically available),
 - không phải là nhất quán tất cả các thời gian (soft-state)
 - nhưng sẽ đạt được trạng thái cuối nhất quán (eventual consistency).
- ☐ Mô hình này rất linh hoạt được cung cấp bởi NoSQL, là cách tiếp cận tương tự để quản lý dữ liệu phi cấu trúc.

NoSQL- So sánh ACID vs BASE

ACID

- ☐ Strong Consistency
- ☐ Isolation
- ☐ Focus on "commit"
- ☐ Nested transactions
- ☐ Less Availability
- ☐ Conservative (pessimistic)
- ☐ Difficult evolution (e.g. schema)

BASE

- ☐ Weak Consistency
- ☐ Availability first
- ☐ Best effort
- ☐ Approximated answers
- ☐ Aggressive (optimistic)
- ☐ Simpler!
- ☐ Faster
- ☐ Easier evolution

NoSQL-Đặc điểm

- ☐ **Mô hình dữ liệu phi quan hệ (Non-relational):** các mô hình này không có mối quan hệ ràng buộc lẫn nhau. Có thể có những cấu trúc dữ liệu phức tạp hơn, nhưng nó không cứng nhắc như mô hình dữ liệu quan hệ. Non-relational là khái niệm không sử dụng các ràng buộc dữ liệu cho nhất quán dữ liệu ở NoSQL database.
- ☐ **Lưu trữ phân tán (Distributed storage):** CSDL NoSQL được phân tán sang nhiều máy tính khác nhau, để cung cấp dữ liệu cho người dùng. Mỗi phần dữ liệu sau đó sẽ được nhân rộng trên một số lượng nhất định máy dự phòng với tính sẵn sàng đáp ứng cao.
- ☐ **Nhất quán cuối (Eventual consistency):** Tính nhất quán của dữ liệu không cần phải đảm bảo ngay tức khắc sau mỗi tác vụ ghi. Một hệ thống phân tán chấp nhận những ảnh hưởng theo phương thức lan truyền và sau một khoảng thời gian (không phải ngay tức khắc), thay đổi sẽ đi đến mọi điểm trong hệ thống, tức là cuối cùng (eventually) dữ liệu trên hệ thống sẽ trở lại trạng thái nhất quán.
- ☐ **Khả năng mở rộng chiều dọc (Vertical scalable):** Khi dữ liệu lớn về lượng, phương pháp tăng cường khả năng lưu trữ và xử lý bằng việc cải tiến phần mềm và cải thiện phần cứng trên một máy tính đơn lẻ được gọi là khả năng mở rộng chiều dọc.
- ☐ **Khả năng mở rộng chiều ngang (Horizontal scalable):** Khi dữ liệu lớn về lượng, phương pháp tăng cường khả năng lưu trữ và xử lý là dùng nhiều máy tính phân tán. Phân tán dữ liệu được hỗ trợ bởi phần mềm tức cơ sở dữ liệu.
- ☐ **Phi cấu trúc (Schema Free):** Không cần thiết kế các bảng, các cấu trúc như của SQL. Thay vào đó chỉ cần lưu trữ các giá trị mới vào CSDL, không cần xác định cấu trúc trước.
- ☐ Không hỗ trợ SQL
- ☐ Dữ liệu lớn (Big Data)
- ☐ Dùng BASE thay thế cho ACID
- ☐ Triển khai linh hoạt (Deployment Flexibility): Việc bổ sung / loại bỏ các node, hệ thống sẽ tự động nhận biết để lưu trữ mà không cần phải can thiệp bằng tay. Hệ thống cũng không đòi hỏi cấu hình phần cứng mạnh, đồng nhất.

3. Ưu điểm và giới hạn NoSQL

- ☐ Các giới hạn của NoSQL

- Sự tin tưởng chưa cao đối với nhiều doanh nghiệp.
- Tính mới mẻ của NoSQL có nghĩa là không có nhiều lập trình viên và người quản trị mà biết công nghệ này.
- Những vấn đề về tính tương thích: Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng (API) riêng của mình, các giao diện truy vấn riêng biệt.
- Khó khăn trong việc lưu trữ các dữ liệu mang nội dung nghiệp vụ phức tạp.

Các ưu điểm của NoSQL [4, 5, 6, 10]

- ☐ Đáp ứng được sự tăng trưởng của dữ liệu lớn.
- ☐ Truy xuất dữ liệu lớn với tốc độ cao.
- ☐ Dữ liệu đa dạng, có cấu trúc, bán cấu trúc hoặc phi cấu trúc.
- ☐ Dữ liệu phức tạp, được lưu trữ và quản lý tại các trung tâm lưu trữ khác.
- ☐ Cần ít tài nguyên và phần cứng của máy chủ.
- ☐ Hỗ trợ chỉ mục tất cả các thuộc tính.
- ☐ Mã nguồn mở.
- ☐ Có thể mở rộng theo chiều dọc.
- ☐ NoSQL được các hãng lớn sử dụng: Các công ty như Amazon, BBC, Facebook và Google dựa vào các CSDL NoSQL.
- ☐ NoSQL và đám mây là một sự trùng khớp tự nhiên, chúng có khả năng tận dụng được việc cung cấp mềm dẻo của đám mây
- ☐ Các CSDL NoSQL hầu hết sử dụng bộ nhớ qua ổ đĩa như là vị trí ghi đầu tiên - vì thế ngăn ngừa được sự thực thi không ổn định của thao tác I/O.

4.Các loại CSDL NoSQL

- ☐ Tính đến thời điểm này có đến 150 loại CSDL NoSQL, chia thành 13 mô hình lưu trữ dữ liệu khác nhau.
- ☐ Có 4 loại CSDL NoSQL thông dụng như:
 - CSDL hướng cột,
 - CSDL cặp khoá – giá trị,
 - CSDL đồ thị,
 - CSDL hướng tài liệu [1].
- ☐ Trong mỗi loại mô hình có các hệ quản trị CSDL đặc trưng. Bảng 2 mô tả phân loại CSDL NoSQL, bảng 3 mô tả sự giống nhau và khác nhau giữa chúng.

Bảng 2. Các loại CSDL NoSQL và hệ quản trị CSDL tương ứng

CSDL	
CSDL cặp khóa – giá trị	Voldemort
	Redis
	Membase
CSDL hướng tài liệu	Riak
	MongoDB
	CouchDB
CSDL hướng cột	Cassandra
	HBase
	Hypertable
CSDL đồ thị	Sesame
	BigData
	Neo4J
	GraphDB
	FlockDB

Cơ sở dữ liệu cặp khoá – giá trị

- Cơ sở dữ liệu cặp khoá – giá trị, trong đó dữ liệu được xác định bằng một khoá duy nhất.
- Khoá (key) là cách duy nhất lấy được giá trị đã lưu trữ. Các giá trị hoàn toàn tách biệt và không phụ thuộc vào nhau, quan hệ của chúng được quản lý bởi các ứng dụng logic.
- Cấu trúc dữ liệu rất đơn giản nên cơ sở dữ liệu cặp khoá – giá trị hoàn toàn không có lược đồ.
- Giá trị mới có thể được thêm vào trong lúc hệ thống đang chạy mà không gây ra bất cứ xung đột dữ liệu nào. Việc sưu tập các cặp khoá – giá trị có thể thêm một số cấu trúc dữ liệu vào mô hình dữ liệu.
- Cơ sở dữ liệu cặp khoá – giá trị hữu ích cho các xử lý đơn giản, chỉ phụ thuộc vào thuộc tính khoá

Cơ sở dữ liệu hướng cột

- Nhìn chung cơ sở dữ liệu hướng cột có nhiều điểm tương đồng với cơ sở dữ liệu quan hệ, nếu nhìn từ bên ngoài, nhưng thật sự có nhiều khác biệt lớn từ bên trong.
- Điểm khác biệt chính nằm ở việc xử lý giá trị “null”. Khi xem xét nhiều trường hợp khác nhau của thuộc tính, cơ sở dữ liệu quan hệ sẽ lưu trữ giá trị “null” trong mỗi cột một tập dữ liệu không có giá trị. Ngược lại, cơ sở dữ liệu hướng cột chỉ chứa một cặp giá trị trong mỗi dòng, nếu bộ dữ liệu cần nó.

- ❑ Ngoài ra, một trong những khác biệt đó chính khác là việc lưu trữ dữ liệu theo dòng (trong cơ sở dữ liệu quan hệ) so với việc lưu trữ dữ liệu theo cột (trong column family databases). Nhưng sự khác biệt lớn là ở chính khái niệm của nó. Chúng ta không thể áp dụng cùng một giải pháp mà chúng ta sử dụng trong cơ sở dữ liệu quan hệ vào trong cơ sở dữ liệu hướng cột. Đó là bởi vì cơ sở dữ liệu cột (column family database) phi quan hệ.

Cơ sở dữ liệu đồ thị

- ❑ Cơ sở dữ liệu đồ thị chuyên dùng trong quản lý dữ liệu hiệu quả với nhiều liên kết.
- ❑ Vì vậy các ứng dụng dựa trên dữ liệu có nhiều mối quan hệ phù hợp với cơ sở dữ liệu đồ thị.
- ❑ Nút (node) và cạnh (edge) bao gồm các đối tượng chứa các cặp khoá – giá trị. Tầm vực của các cặp khoá – giá trị được định nghĩa trong lược đồ, nên các ràng buộc phức tạp được mô tả dễ dàng.
- ❑ Thuộc tính của đồ thị được xác định bởi các framework của các hệ quản trị cơ sở dữ liệu khác nhau.

Cơ sở dữ liệu hướng văn bản

- ❑ Là một chương trình được thiết kế dùng để lưu trữ, truy xuất và quản lý hướng tài liệu hay dữ liệu bán cấu trúc hoặc thông tin.
- ❑ Trái ngược lại với cơ sở dữ liệu quan hệ truyền thống, khái niệm về quan hệ (relations hay bảng) trong những hệ thống này được thiết kế xung quanh một khái niệm trừu tượng gọi là tài liệu (document).
- ❑ Một trong các đặc điểm khác biệt của một cơ sở dữ liệu hướng tài liệu là việc sử dụng key-document (hoặc key-value) để lấy một tài liệu, cơ sở dữ liệu sẽ cung cấp một API hoặc ngôn ngữ truy vấn cho phép bạn lấy các tài liệu dựa trên nội dung
- ❑ Cơ sở dữ liệu hướng tài liệu đóng gói các cặp khoá – giá trị trong như tài liệu, và trong các tài liệu này khoá là định danh duy nhất gọi là _id, dùng xác định các tài liệu một cách rõ ràng.
- ❑ Ngược với cơ sở dữ liệu cặp khoá – giá trị, giá trị tại đây không bị che khuất nên có thể dễ dàng truy vấn tốt. Do đó, các cấu trúc dữ liệu phức tạp như các đối tượng lồng nhau có thể được xử lý thuận tiện.
- ❑ Tương tự như cơ sở dữ liệu cặp khoá – giá trị. Cơ sở dữ liệu hướng tài liệu không có bất kỳ hạn chế nào về mô hình dữ liệu. Việc lưu trữ các document mới bao gồm các thuộc tính dễ dàng được thực hiện như thêm các thuộc tính mới vào các document sẵn có trong lúc chạy. Cơ sở dữ liệu hướng tài liệu cung cấp tra cứu đa thuộc tính trên các dòng dữ liệu, bao gồm các cặp khoá – giá trị khác nhau.

Bảng 3. Sự giống và khác nhau giữa 4 loại CSDL NoSQL

Loại CSDL NoSQL		Khả năng truy vấn		Quản lý đồng thời		Phân vùng		Nhân bản	
		Java API	Ngôn ngữ truy vấn	Khóa	Khóa tích cực	Dây cơ sở	Bảng băm	Đọc	Ghi
Cặp khóa – giá trị	Redis	Có	Không	Không	Có	Không	Có	Không	Không
	Membase	Có	Không	Không	Có	Không	Có	Không	Không
Hướng tài liệu	MongoDB	Có	Không	Không	Không	Có	Không	Không	Không
	CouchDB	Có	Không	Không	Không	Không	Có	Có	Có
Hướng cột	Cassandra	Có	Có	Không	Không	Không	Có	Không	Không
	Hbase	Có	Có	Có	Không	Có	Không	Không	Không
Đồ thị	Neo4j	Có	Có	Có	Không	Không	Không	Có	Có
	Graph DB	Có	Không	Không	Không	Không	Không	Không	Không

10. các cú pháp tạo role, user, phân x1,x2 thuộc a1, a2

1. Role

- Thêm một role

Hàm:

Create ROLE Ten-role

Hoặc

Exec sp_addrole 'Tên-role'

Ví dụ:

Use QLSV

exec sp_Addrole 'KeToan' --Thêm 1 role có tên kế toán trên CSDL là QLSV

- Xóa role

DROP ROLE Ten-role

Hoặc

Exec sp_droprole 'Ten-role'

2. User

- Thêm user

Create USER Ten-user [for Ten-login]

Hoặc

Exec sp_adduser 'Ten-login', 'ten-user', 'ten-role' - **Xóa user**

DROP USER Ten-user

Hoặc

Exec sp_dropuser 'Ten-user'

- Chia các user vào các role:

sp_addrolemember 'tên-role', 'security_account hay ten-user'

3. Phân quyền

- Cú pháp cấp quyền:
GRANT <some permission> ON <some object>

TO <some user, login, or group>

Cú pháp từ chối quyền:
DENY <some permission> ON <some object>

TO <some user, login, or group>

- Cú pháp thu hồi/xóa quyền:

REVOKE <some permission> ON <some object> FROM/TO <some user, login, or group>

-----TAO 3 ROLE TU R1->R3-----

```
Exec sp_addrole 'r1'  
Exec sp_addrole 'r2'  
Exec sp_addrole 'r3'
```

```
--r3 có quyền select, insert, delete trên table SACH
GRANT SELECT, INSERT, DELETE ON SACH TO r3

--r1 có quyền select, delete trên Table THELOAI, SACH
GRANT SELECT,DELETE ON THELOAI TO R1
GRANT SELECT,DELETE ON SACH TO R1

-- R2 có quyền update, delete trên Table SACH
GRANT UPDATE,DELETE ON SACH TO R2

-- r1 bị từ chối quyền insert trên THELOAI,SACH
DENY INSERT ON THELOAI TO R1
DENY INSERT ON SACH TO R1

-- R2 bị từ chối quyền delete trên table THANHVIEN
DENY DELETE ON THANHVIEN TO R2

--R3 bị từ chối quyền update trên table SACH
DENY UPDATE ON SACH TO R3

--R3 bị thu hồi quyền insert trên table SACH
REVOKE INSERT ON SACH TO R3

--R1 bị thu hồi quyền delete trên table THELOAI
REVOKE DELETE ON THELOAI TO R1
```

-----TAO 3 USER TU U1->U3-----

```
= CREATE LOGIN ITACHI WITH PASSWORD='123456',  
  DEFAULT_DATABASE=QLTV  
  CREATE USER U1 FOR LOGIN ITACHI
```

```
= CREATE LOGIN ITACHI1 WITH PASSWORD='',  
  DEFAULT_DATABASE=QLTV  
  CREATE USER U2 FOR LOGIN ITACHI1
```

```
= CREATE LOGIN ITACHI2 WITH PASSWORD='',  
  DEFAULT_DATABASE=QLTV  
  CREATE USER U3 FOR LOGIN ITACHI2
```

-----TAO NHOM: U1 THUOC R1; U2 THUOC R2; U3 THUOC R3-----

```
Exec sp_addrolemember 'r1', 'U1'  
Exec sp_addrolemember 'r2', 'U2'  
Exec sp_addrolemember 'r3', 'U3'
```

QuanLyThuVien.sql...647\MR Itachi (54))* X

```
-- U1 có quyền select, delete trên Table THELOAI, SACH
GRANT SELECT,DELETE ON THELOAI TO U1
GRANT SELECT, DELETE ON SACH TO U1

-- U2 có quyền update, delete trên Table SACH
GRANT UPDATE, DELETE ON SACH TO U2

-- U3 có quyền insert trên Table THELOAI, SACH, THANHVIEN
GRANT INSERT ON THELOAI TO U3
GRANT INSERT ON SACH TO U3
GRANT INSERT ON THANHVIEN TO U3

-- U1 bị từ chối quyền insert trên THELOAI, SACH
DENY INSERT ON THELOAI TO U1
DENY INSERT ON SACH TO U1
```

QuanLyThuVien.sql...647\MR Itachi (54))* X

```
-- U2 bị từ chối quyền delete trên table THANHVIEN
DENY DELETE ON THANHVIEN TO U2

--U3 Có quyền select va delete tren table SACH, THANHVIEN
GRANT SELECT, DELETE ON SACH TO U3
GRANT SELECT, DELETE ON THANHVIEN TO U3

--U1 bị thu hồi quyền select trên table THELOAI
REVOKE SELECT ON THELOAI TO U1

--U3 bị thu hồi quyền DELETE trên table THANHVIEN
REVOKE DELETE ON THANHVIEN TO U3
```

11. phục hồi csdl bị mất

Các chế độ khôi phục (Recovery Models)

- **Chế độ Full Recovery:** Đây là chế độ cho phép phục hồi dữ liệu với ít rủi ro nhất. Nếu một database ở trong chế độ này thì tất cả các hoạt động không chỉ insert, update, delete mà kể cả

insert bằng **Bulk Insert**, hay **bcp** đều được log vào file transaction log. Khi có sự cố thì ta có thể phục hồi lại dữ liệu ngược trở lại tới một thời điểm trong quá khứ. Khi file dữ liệu bị hư nếu ta có thể sao lưu được file transaction log thì ta có thể phục hồi CSDL đến thời điểm transaction gần nhất đã được xác nhận (committed).

- **Chế độ Bulk-Logged Recovery:** Ở chế độ này các hoạt động mang tính hàng loạt như Bulk Insert, bcp, Create Index, WriteText, UpdateText chỉ được log minimum vào File Transaction Log đủ để cho biết là các hoạt động này có diễn ra mà không log toàn bộ chi tiết như trong chế độ Full Recovery. Các hoạt động khác như Insert, Update, Delete vẫn được log đầy đủ để dùng cho việc phục hồi sau này.
- **Chế độ Simple Recovery:** Ở chế độ này thì File Transaction Log được cắt xén thường xuyên và không cần sao lưu. Với chế độ này bạn chỉ có thể phục hồi tới thời điểm backup gần nhất mà không thể phục hồi tới một thời điểm trong quá khứ.

12 . NoSQL và SQL

1.thông tin là gì

Thông tin là bất kỳ thực thể hoặc hình thức cung cấp câu trả lời cho một câu hỏi nào hoặc giải quyết sự không chắc chắn.

Do đó, nó liên quan đến **dữ liệu** và **kiến thức**, vì dữ liệu đại diện cho các giá trị được gán cho các tham số và kiến thức biểu thị sự hiểu biết về những điều thực tế hay khái niệm trừu tượng.^[1] Vì nó liên quan đến dữ liệu, sự tồn tại của thông tin không nhất thiết phải kết hợp với một người quan sát (ví dụ như nó tồn tại ngoài **chân trời sự kiện**), trong trường hợp hiểu biết thì thông tin yêu cầu một người quan sát có nhận thức.

2.Dữ liệu là gì

Nói chung, **dữ liệu** bao gồm những **mệnh đề** phản ánh thực tại. Một phân loại lớn của các mệnh đề quan trọng trong thực tiễn là các **đo đạc** hay **quan sát** về một đại lượng biến đổi. Các mệnh đề đó có thể bao gồm các **số**, **từ** hoặc **hình ảnh**.

Dữ liệu là thông tin dưới dạng **ký hiệu, chữ viết, chữ số, hình ảnh, âm thanh** hoặc dạng tương tự.

3. Quản lý thông tin (QLTT) là gì?

Quản lý thông tin là gì?

Thiết kế tổ chức thông tin

Xử lý thông tin

Mục tiêu QLTT

Giảm nhu cầu xử lý thông tin (thừa)

Tăng năng lực tổ chức, xử lý thông tin

Quản lý dữ liệu là gì?

Quản lý dữ liệu là tập các chức năng: phát triển, thực hiện và giám sát kế hoạch, chính sách, chương trình và hoạt động nhằm kiểm soát, bảo vệ, cung cấp và nâng cao giá trị của dữ liệu”

4. Tổ chức thông tin

☐ **TCTT gồm các hoạt động:**

Mô tả tài liệu, lập chỉ mục và phân loại

Được thực hiện trong các thư viện, cơ sở dữ liệu, vv

Những hoạt động này được thực hiện bởi cán bộ thư viện, các chuyên gia, cũng như bởi các thuật toán máy tính.

5. Xử lý thông tin

☐ **Định nghĩa 1.**

Là các ngành khoa học liên quan đến việc thu thập, thao tác, lưu trữ, tìm kiếm, và phân loại thông tin đã ghi nhận.

6. Sự phức tạp quản lý thông tin

- ☐ Thiếu các nguồn lực để tổ chức, quản lý và cải thiện hệ thống thông tin.
- ☐ Thiếu định nghĩa với nhiều thông tin
- ☐ Ít hoặc không có sự đánh giá và hỗ trợ quản lý thông tin của các quản lý cấp cao.
- ☐ Sự kém chất lượng của thông tin, vd: sự thiếu nhất quán, sự lặp lại
- ☐ Hệ thống đã lỗi thời cần thay thế hoặc nâng cấp.
- ☐ Số lượng lớn các hệ thống quản lý thông tin phân tán ít liên kết nhau
- ☐ Thiếu đồng bộ giữa các hệ thống thông tin.
- ☐ Cạnh tranh giữa các hệ thống quản lý thông tin.
- ☐ Thiếu con đường chiến lược về quản lý hệ thống thông tin.
- ☐ Thiếu sự rõ ràng của các chiến lược và mục tiêu của tổ chức.
- ☐ Có khác nhau về các vấn đề đa dạng và nhu cầu chưa được giải quyết.
- ☐ Nhân viên hạn chế cập nhật hệ thống thông tin hiện có.
- ☐ Cản trở trong việc chuyển đổi tập quán làm việc cũ và các quá trình của nhân viên.

7. Một số thuộc tính quan trọng của dữ liệu

- ☐ Tính liên quan (relevance): tính hữu ích của các dữ liệu trong bối cảnh công việc.
- ☐ Rõ ràng (Clarity): sự sẵn có của một định nghĩa rõ ràng cho các dữ liệu.
- ☐ Thống nhất (Consistency): sự phù hợp của cùng một loại dữ liệu từ các nguồn khác nhau.

- ☐ Kịp thời: sự sẵn có của dữ liệu vào thời điểm cần thiết và làm thế nào cho đến ngày đó là dữ liệu.
- ☐ Độ chính xác (Accuracy): làm thế nào gần với sự thật là các dữ liệu.
- ☐ Hoàn chỉnh (Completeness): bao nhiêu dữ liệu cần thiết có sẵn.
- ☐ Khả năng tiếp cận (Accessibility): ở đâu, như thế nào, và ai là người có trách nhiệm.
- ☐ Chi phí (Cost): chi phí phát sinh trong việc thu thập các dữ liệu, và làm cho nó có sẵn để sử dụng.
- ☐ Kích thước
- ☐ Loại dữ liệu
- ☐ Ngày tạo, ngày cập nhật, ngày truy xuất
- ☐ Chủ sở hữu
- ☐ Dữ liệu định tính
- ☐ Dữ liệu định lượng

8. Dữ liệu vs Thông tin

- ☐ Dữ liệu
- Sự kiện
- Không nội dung
- Con số, hình ảnh, âm thanh, văn bản
- ☐ Thông tin
- Dữ liệu có nội dung
- Dữ liệu đã xử lý
- Thêm giá trị vào dữ liệu nhờ
- Phân tích
- Tổng hợp
- Tổ chức lại

9. Phân loại hệ CSDL

- ☐ Dựa theo mô hình :Quan hệ ; Mạng ; Cây ; Đối tượng
- ☐ Dựa vào số người dùng :Single user; Multiuser
- ☐ Dựa số các site chứa sự phân bố dữ liệu :Tập trung (centralized DB) ; Phân tán (distributed DB)
- ☐ Tính đồng nhất của các nguồn dữ liệu :Đồng nhất ; Không đồng nhất
- ☐ Dựa vào chi phí:Thấp [100, 10000]; Trung bình [10000, 100000]; Lớn >100000USD
- ☐ Mục đích :Tổng quát; Chuyên biệt

10. Phân tích dữ liệu

- ☐ Phân tích các dữ liệu là một quá trình:
- Kiểm tra, dọn dẹp, chuyển, và mô hình hóa dữ liệu
- ☐ Mục tiêu

Khám phá thông tin hữu ích,
Tìm kết luận, và hỗ trợ ra quyết định.

☐ Phân tích dữ liệu có nhiều khía cạnh và phương pháp tiếp cận, bao gồm các kỹ thuật đa dạng: Khoa học kỹ thuật, và các lĩnh vực khoa học xã hội.

11. Thu thập dữ liệu (Data collection)

- ☐ Dữ liệu được thu thập từ nhiều nguồn khác nhau.
- ☐ Các yêu cầu được đặt ra bởi các nhà phân tích, khách hàng.
- ☐ Các dữ liệu cũng có thể được thu thập từ các cảm biến trong môi trường (camera giao thông, vệ tinh, thiết bị ghi âm, vv)
- ☐ Có thể thu được thông qua các cuộc phỏng vấn, tải từ các nguồn trực tuyến, hoặc tài liệu đọc.

12. Xử lý dữ liệu (Data Process)

- ☐ Gồm các giai đoạn khôn ngoan được sử dụng để chuyển đổi các DL thô thành DL tinh hay kiến thức.
- ☐ Dữ liệu ban đầu thu được phải được xử lý hoặc tổ chức để phân tích.
- ☐ Ví dụ, đưa dữ liệu thành các hàng và cột trong một bảng để phân tích thêm, nhờ các công cụ có sẵn.

13. Làm sạch dữ liệu (Data clean)

☐ Sau khi xử lý và tổ chức, các dữ liệu có thể: không đầy đủ, chứa bản sao, hoặc có sai sót.

- ☐ Sự cần thiết phải làm sạch dữ liệu sẽ phát sinh từ phương cách mà dữ liệu đã được nhập và lưu trữ.
- ☐ Làm sạch dữ liệu là quá trình ngăn ngừa và sửa chữa các lỗi này.

14. Phân tích dữ liệu

- ☐ Một khi dữ liệu được làm sạch, nó có thể được phân tích.
- ☐ Các nhà phân tích có thể áp dụng một loạt các kỹ thuật phân tích dữ liệu được gọi là thăm dò để bắt đầu tìm hiểu các thông điệp chứa trong dữ liệu.
- ☐ Quá trình thăm dò có thể dẫn đến làm sạch dữ liệu bổ sung hoặc yêu cầu bổ sung cho dữ liệu, do đó, những hoạt động có thể được lặp đi lặp lại trong tự nhiên.
- ☐ PP thống kê có thể được tạo ra để giúp hiểu được dữ liệu (sum, max, min, avg, count. . .)
- ☐ Trực quan dữ liệu cũng có thể được sử dụng để kiểm tra dữ liệu ở định dạng đồ họa, để có được cái nhìn sâu sắc thêm về các thông điệp trong dữ liệu.

15. Khôi phục dữ liệu (Data recovery)

- ☐ Khôi phục dữ liệu
- ☐ Khôi phục CSDL
- ☐ Phục hồi dữ liệu là quá trình cứu vớt (salvage) và xử lý các dữ liệu từ các dữ liệu bị hư hỏng, hoặc các phương tiện lưu trữ thứ cấp không thể tiếp cận bình thường.
- ☐ Một số tình huống đặc biệt
 - ☐ Khôi phục dữ liệu được hiểu là: chuyển dữ liệu từ mã hóa đến không mã; từ ẩn đến không ẩn.
- ☐ Có 3 kịch bản thường gặp
 - ☐ Lỗi HĐH.
 - ☐ Lỗi bộ nhớ thứ cấp
 - ☐ Bị xóa nhầm
- ☐ Lỗi HĐH (lỗi logic)
 - ☐ Sao chép các dữ liệu quan trọng trên một thiết bị lưu trữ khác trước khi có sự cố
 - ☐ Nếu xảy ra sự cố, có thể khôi phục dữ liệu như phần mềm chuyên biệt: Testdisk. . .
- ☐ Lỗi đĩa (lỗi physic)
 - ☐ Do nhiều nguyên nhân: trầy trên bề mặt, hỏng động cơ.
 - ☐ Dữ liệu bị mất ít hoặc nhiều.
 - ☐ Phụ thuộc tình hình mà dùng các phần mềm khôi phục dữ liệu
 - ☐ Nên vận hành máy tính đúng qui tắc để tránh thiệt hại ở dạng này.
- ☐ Xóa nhầm
 - ☐ Thông thường các files chỉ bị đánh dấu xóa chứ không xóa thực sự trong thời gian đầu. Một số công cụ có thể hỗ trợ khôi phục trong tình huống này.
 - ☐ Nếu file xóa đã bị ghi chồng lên (overwrite) thì việc khôi phục là không thể hoặc rất khó khăn
- ☐ Khôi phục CSDL là đưa CSDL về trạng thái nhất quán gần nhất, nghĩa là khôi phục về trạng thái đúng gần nhất ở quá khứ.
- ☐ Để có thể khôi phục được cần giữ các thông tin làm thay đổi dữ liệu trong thời gian thực hiện giao tác (transaction), các thông tin này nằm ngoài CSDL, nó được ghi trong Systemlog.
- ☐ Systemlog: file nhật kí hệ thống sử dụng cho việc khôi phục
- ☐ Chiến thuật
 - ☐ CSDL hỏng nặng: sử dụng file backup gần nhất để khôi phục
 - ☐ CSDL hỏng nhẹ: dùng cơ chế UNDO và REDO

16. Transaction.

- ☐ Giao tác là tập các hành vi được định nghĩa như một đơn vị nhỏ nhất.
- ☐ Một transaction chỉ có thể tồn tại:

Thành công

Không làm gì (All or nothing)

- ☐ Một giao tác thất bại thì cập nhật CSDL buộc phải rollback , tất cả dữ liệu bị thay đổi bởi giao tác cần khôi phục về giá trị trước đó (before value).
- ☐ Một giao tác T1 bị rollback có thể kéo theo T2, T3, ... bị rollback, do T2,T3, ... đọc các dữ liệu cần bị undo của T1. Các trans T2, T3 ... rollback kéo theo T1',T2'... rollback theo cũng do nguyên nhân giống trên và chúng tạo thành hiện tượng cascade rollback.

17. Sao lưu dữ liệu (Backup)

- ☐ Trong CNTT, một bản sao lưu, hoặc quá trình sao lưu, nghĩa là đề cập đến việc sao chép và lưu trữ dữ liệu máy tính và sử dụng nó để khôi phục lại dữ liệu ban đầu sau khi dữ liệu hư hay mất.

18. Kho dữ liệu lưu trữ (Repository)

- ☐ Bất kỳ chiến lược sao lưu được bắt đầu với một khái niệm về một kho lưu trữ dữ liệu.

Kho không có cấu trúc

Kho đầy đủ hệ thống

Kho dạng Incremental

Kho dạng differential

- ☐ Kho không có cấu trúc

Một kho lưu trữ không cấu trúc có thể chỉ đơn giản là một chồng CD-R, DVD-R, USB. . . với các thông tin tối thiểu về những gì đã được sao lưu và khi nào.

Dễ để thực hiện

Ít có khả năng để đạt được một mức độ cao về khả năng thu hồi.

- ☐ Kho đầy đủ hệ thống

Kho thuộc loại này có chứa hình ảnh hoàn chỉnh của hệ thống được chụp tại một hay nhiều điểm trong thời gian

- ☐ Kho dạng Incremental

lưu trữ các bản sao từ nhiều điểm theo thời gian bằng cách tổ chức các dữ liệu thành số gia của các biến đổi.

Giúp loại bỏ sự cần thiết để lưu trữ các bản sao của dữ liệu không thay đổi.

Cần nhiều bộ dữ liệu khi khôi phục

- ☐ Kho dạng differential

Mỗi sao lưu dạng này chỉ lưu các dữ liệu đã thay đổi kể từ khi sao lưu cuối cùng đầy đủ.

ưu điểm: chỉ có tối đa hai bộ dữ liệu cần thiết khi khôi phục lại dữ liệu.

19. Chọn thiết bị sao lưu

- ☐ HardDisk trên máy

Đơn giản

Thời gian backup nhỏ

Tốn chi phí bộ nhớ

Rủi ro: mất, hỏng, virus. . .

- ☐ Thiết bị lưu trữ khác: CD, USB

Thời gian backup nhiều hơn

Hạn chế rủi ro đồng thời

- ☐ Cold database backup

Cơ sở dữ liệu được đóng hoặc bị khóa và không có sẵn cho người dùng. Các datafiles không thay đổi trong suốt quá trình dự phòng để các cơ sở dữ liệu có trạng thái thống nhất khi nó hoạt động bình thường.

- ☐ Cold database backup

Một số hệ thống quản lý cơ sở dữ liệu cung cấp một phương tiện để tạo ra một hình ảnh sao lưu cơ sở dữ liệu trong khi nó là trực tuyến và sử dụng được.

20. Crystal Report

- ☐ Crystal Report là công cụ thiết kế báo cáo cho phép tạo ra những báo cáo (từ đơn giản đến phức tạp) bằng cách tìm và định dạng dữ liệu từ một hay nhiều nguồn dữ liệu khác nhau.
- ☐ Hỗ trợ các chức năng in ấn, kết xuất sang các định dạng khác: PDF, Excel, Word...

21. 1.Các mô hình dữ liệu

- ☐ Mô hình là gì?

Là thuật ngữ để biểu diễn các hiện tượng trong một phương thức dễ đọc.

Mô hình cũng có thể là sự trừu tượng hóa, đơn giản hóa về một thế giới thực, là cầu nối giữa lý thuyết và thực tiễn.

- ☐ Mô hình có thể được chia thành hai loại:

Số hóa và không số hóa.

Mô hình không số hóa thì dễ hiểu nhưng khó quản trị bởi máy tính.

- ☐ Mô hình dữ liệu

Là phương thức biểu diễn lại dữ liệu sao cho thuận lợi để biểu diễn trên máy tính. Mô hình dữ liệu bao gồm:

- ☐ Hệ thống ký hiệu để mô tả dữ liệu
- ☐ Tập hợp các phép toán thao tác trên dữ liệu đó

21.2 Các mô hình dữ liệu

- ☐ Mô hình dữ liệu phẳng (flat model)
- ☐ Mô hình phân cấp (Hierarchical model)
- ☐ Mô hình mạng (Network model)
- ☐ Mô hình dữ liệu quan hệ (Relational Model)
- ☐ Mô hình dữ liệu hướng đối tượng (Object Oriented model)
- ☐ Mô hình dữ liệu quan hệ đối tượng (Object-relational model)
- ☐ Mô hình dữ liệu phẳng
 - ☐ Một cơ sở dữ liệu phẳng là một hệ thống cơ sở dữ liệu đơn giản, trong đó mỗi cơ sở dữ liệu được biểu diễn như là một bảng duy nhất
 - ☐ Tuy nhiên, một số nhà phát triển ứng dụng vẫn sử dụng các tập tin phẳng để giảm chi phí và tính phức tạp của việc tích hợp cơ sở dữ liệu quan hệ.
 - ☐ Cơ sở dữ liệu phẳng đôi khi cũng được gọi là cơ sở dữ liệu tệp phẳng (flat-file databases).

22. Đặc điểm các HQTCSDL

- ☐ Mô hình phân cấp (Hierarchical model)

Đưa ra vào những năm 60

Dữ liệu được tổ chức thành cấu trúc cây

Các nút (node) là tập các thực thể

Các cạnh là các mối quan hệ giữa hai nút theo mối quan hệ nhất định. Hay nói cách khác:

- ☐ Là mô hình dữ liệu trong đó các bản ghi được sắp xếp theo cấu trúc top-down(tree).
- ☐ Một con chỉ có một cha, chỉ có một đường truy nhập tới dữ liệu đó trước.

CSDL tập tin (Flat Files)

Thuận lợi	Khó khăn
Nhiều phương thức truy xuất, e.g., tuần tự, không tuần tự, chỉ mục	<ul style="list-style-type: none">▪ Dữ liệu trùng lặp▪ Cần chi phí xử lý để dữ liệu đồng nhất, điều khiển việc truy xuất▪ Bảo mật kém▪ Độ mịn của chia sẻ đơn vị dữ liệu thô▪ Rất khó trong vấn đề điều khiển việc nhiều chương trình, nhiều user cùng truy xuất dữ liệu

☐ Ưu điểm

đơn giản về khái niệm

nhóm dữ liệu có thể được liên quan đến nhau

dữ liệu liên quan có thể được xem với nhau

Tập trung dữ liệu

giảm sự dư thừa và phát huy tính nhất quán

☐ Nhược điểm

Biểu diễn hạn chế về mối quan hệ dữ liệu

không cho phép quan hệ(M: N)

Thực hiện phức tạp

cần kiến thức chuyên sâu về lưu trữ dữ liệu vật lý

Sự phụ thuộc cấu trúc

truy cập dữ liệu đòi hỏi đường dẫn lưu trữ vật lý

Thiếu tiêu chuẩn

23. Mô hình mạng (Network model)

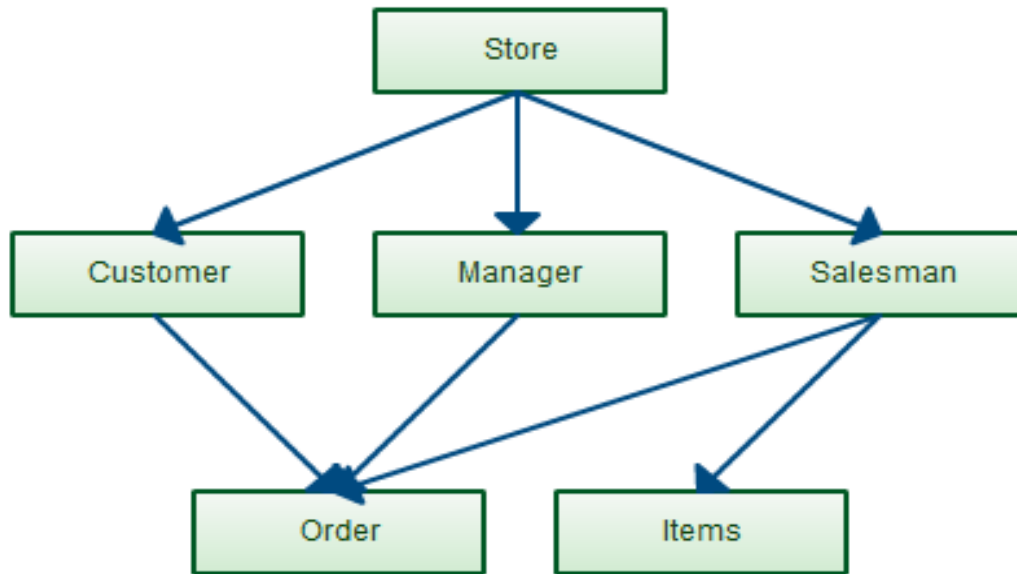
☐ Được đưa vào cuối những năm 60.

☐ Dữ liệu được tổ chức thành một đồ thị có hướng

☐ Các đỉnh là các thực thể

☐ Các cung là quan hệ giữa hai đỉnh, một kiểu bản ghi có thể liên kết với nhiều kiểu bản ghi khác.

☐ Một con có thể có nhiều cha, có nhiều đường truy nhập đến một dữ liệu cho trước



☐ Ưu điểm

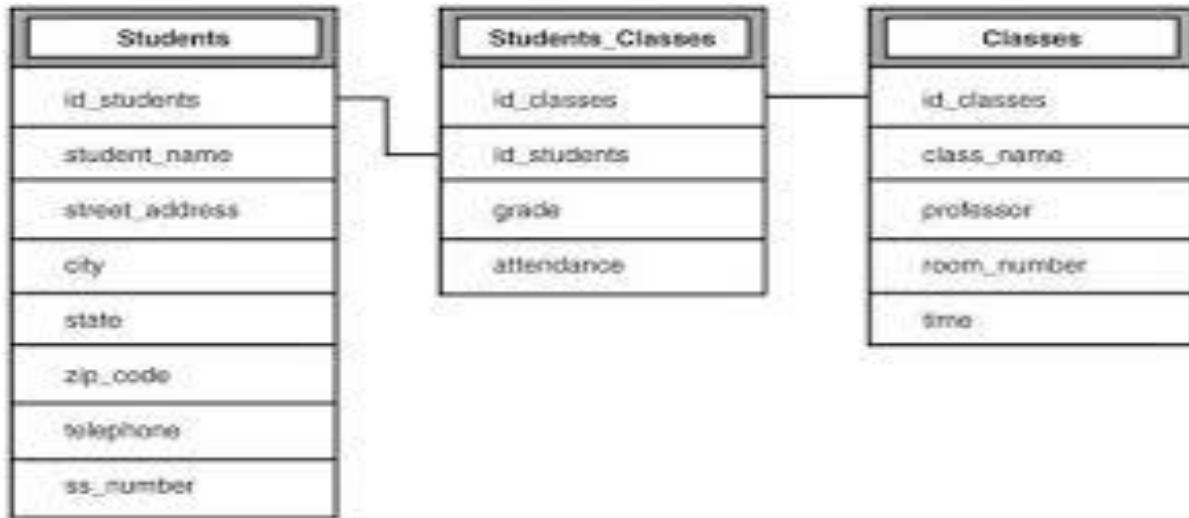
Cho phép nhiều loại quan hệ dữ liệu
 Hiệu quả và truy cập dữ liệu linh hoạt
 Phù hợp với các tiêu chuẩn
 tăng cường quản trị cơ sở dữ liệu

☐ Nhược điểm

Hệ thống phức tạp
 Đòi hỏi phải có sự quen thuộc với cấu trúc nội bộ để truy cập dữ liệu
 Thay đổi cấu trúc nhỏ đòi hỏi thay đổi chương trình.

24. Mô hình dữ liệu quan hệ (Relational Model)

- ☐ E.F Codd đưa vào đầu những năm 70
- ☐ Dựa trên lý thuyết tập hợp và đại số quan hệ (*chương sau*).
- ☐ Vì tính chất chặt chẽ của toán học về lý thuyết tập hợp nên mô hình này đã mô tả dữ liệu một cách rõ ràng, mềm dẻo và là mô hình thông dụng.
- ☐ Nhiều hệ QTCSDL đều tổ chức dữ liệu theo mô hình dữ liệu quan hệ.
- ☐ Trong đó dữ liệu được tổ chức dưới dạng bảng, các phép toán thao tác trên dữ liệu dựa trên lý thuyết tập hợp của toán học.



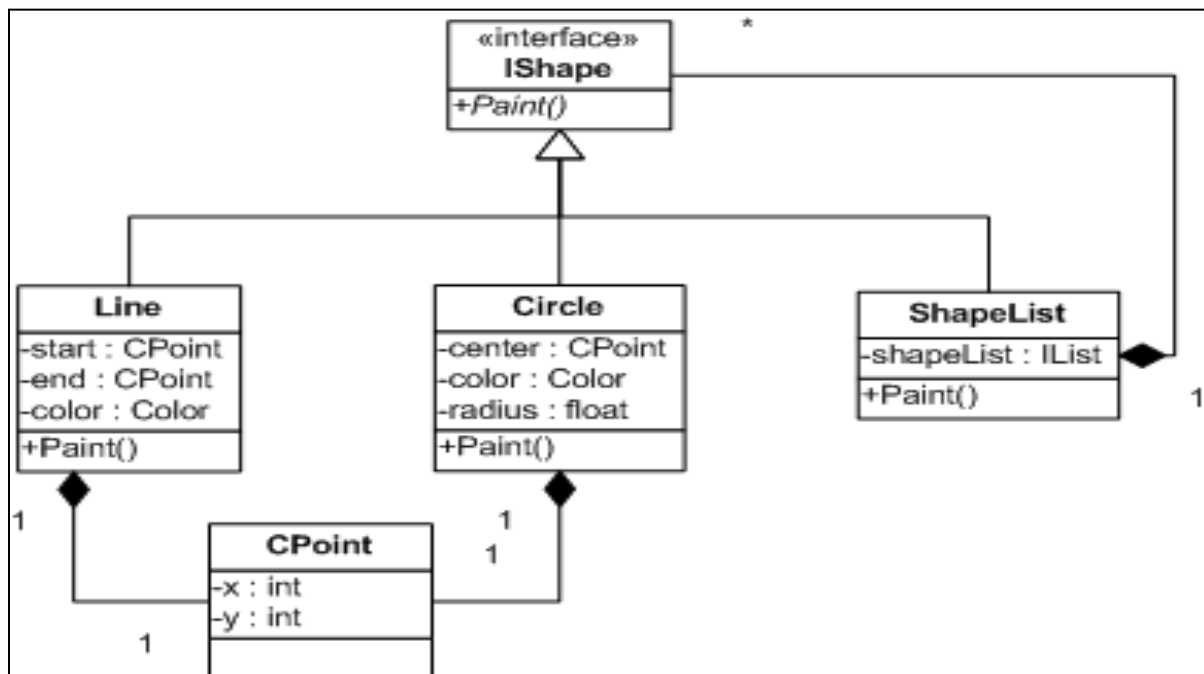
25. Mô hình dữ liệu hướng đối tượng (Object Oriented model)

- ☐ Trong đó các thuộc tính dữ liệu và các phương thức thao tác trên các thuộc tính đó được đóng gói trong các cấu trúc gọi là đối tượng.
- ☐ Hình thành nhiều khái niệm mới

Kế thừa

Đa hình

Đóng gói ...



CSDL quan hệ (Relational Database)

Thuận lợi	Khó khăn
Dễ hiểu	<ul style="list-style-type: none">▪ Dùng nhiều quan hệ để biểu diễn một thể hiện trong thế giới thực▪ Hạn chế về kiểu dữ liệu

CSDL hướng đối tượng (Object Oriented Database)

Thuận lợi	Khó khăn
<ul style="list-style-type: none">▪ Rút gọn số lượng lớn các quan hệ▪ Nhờ inheritance và polymorphism có thể xử lý các đối tượng có phương thức khó.▪ Chấp nhận các kiểu dữ liệu khác nhau hơn.	<ul style="list-style-type: none">▪ Chuyển đổi từ CSDLQH đến nó là khó khăn▪ Một hệ QTCSDL thường đi kèm với ngôn ngữ lập trình, nên kém linh hoạt▪ Tạo ra vài vấn đề khi xóa dữ liệu lớn▪ Không thể truy vấn trên 2 lớp như CSDL quan hệ

26. Mô hình dữ liệu quan hệ đối tượng

- ☐ Là mô hình lai giữa 2 mô hình Quan hệ và hướng đối tượng
- ☐ Tận dụng ưu điểm và hạn chế nhược điểm của 2 mô hình trên.

CSDL đối tượng quan hệ (Object Relational Database)

Thuận lợi	Khó khăn
Khả năng lưu trữ lớn Tốc độ truy xuất cao	<ul style="list-style-type: none">▪ Kiến trúc không tương thích các ứng dụng WEB

27. NoSQL

- ☐ NoSQL không có nghĩa "không có sql". Nó là dữ liệu không quan hệ.
- ☐ Một cơ sở dữ liệu NoSQL cung cấp một cơ chế lưu trữ và truy xuất dữ liệu mà tính nhất quán lỏng lẻo hơn so với cơ sở dữ liệu quan hệ (dữ liệu không cần phải đảm bảo ngay tức khắc sau mỗi phép ghi)
- ☐ Cơ sở dữ liệu NoSQL có đặc trưng:
 - ☐ Không quan hệ
 - ☐ Phân tán
 - ☐ Mã nguồn mở
 - ☐ Có khả năng mở rộng theo chiều ngang
 - ☐ Phù hợp cho cộng đồng dùng điện toán đám mây.
 - ☐ Hiệu suất cao là tiêu chí quan trọng.
 - ☐ Kích thước lớn
 - ☐ Chủ yếu là truy vấn, rất ít cập nhật (update)
 - ☐ Tính nguyên tố của một giao tác không cần thiết
 - ☐ Trả lời nhanh là quan trọng hơn chính xác

28. Mô hình là gì?

- ☐ Mô hình là thuật ngữ để biểu diễn các hiện tượng trong một phương thức dễ đọc. Mô hình cũng có thể là sự trừu tượng hóa, đơn giản hóa về một thế giới thực, là cầu nối giữa lý thuyết và thực tiễn. Mô hình có thể được chia thành hai loại: số hóa và không số hóa.

29. Mô hình dữ liệu là gì?

- ☐ Mô hình dữ liệu là phương thức biểu diễn phần dữ liệu của thế giới thực một cách dễ hiểu đối với máy tính. Các mô hình dữ liệu quen thuộc gồm: mô hình quan hệ, mô hình thực thể kết hợp, mô hình hướng đối tượng

30. Mô hình hóa là gì?

Mô hình hóa là một phương tiện giúp đơn giản hóa thế giới thực bằng các mô hình.
Mô hình hóa giúp con người hiểu rõ hơn về hệ thống dưới một góc nhìn.

31. Mô hình hóa dữ liệu là gì?

Mô hình hóa dữ liệu là mô hình chuẩn được cung cấp tạo thuận lợi cho việc tổ chức dữ liệu sao cho việc truy vấn dữ liệu là dễ dàng.

32. Mô hình quan niệm là gì?

- ☐ Mô hình quan niệm là một mô hình thực hiện của các thành phần của khái niệm, được sử dụng để giúp mọi người biết, hiểu, hoặc mô phỏng các đối tượng trên mô hình đại diện.

33. Mô hình dữ liệu quan niệm là gì?

Một mô hình dữ liệu quan niệm niệm là hình thức trừu tượng nhất của mô hình dữ liệu. Nó là hữu ích để giao tiếp các ý tưởng, vì tính đơn giản của nó.

34. 2.Mô hình ERD

- ☐ Đặt vấn đề
- ☐ Các yếu tố cơ bản

Thực thể

Bản số

Mối kết hợp

- ☐ Chuyển mô hình dữ liệu từ mức quan niệm sang mức logic
- ☐ Mô hình dữ liệu mức đơn giản (quan niệm) thuộc công đoạn phân tích.
- ☐ Từ 1 bài toán mô tả bằng văn bản và ngôn ngữ tự nhiên với đặc điểm:

Không thứ tự.

Lẫn lộn giữa các khái niệm.

Thiếu vừa thừa.

Pha trộn giữa dữ liệu và xử lý không rõ ràng, không thứ tự.

- ☐ Phân tích dữ liệu chuyển sang 1 bài toán:

có thứ tự, đúng, đủ, tách dữ liệu và xử lý, định nghĩa lại khái niệm.

Đồng thời mô tả phần dữ liệu dưới dạng ngôn ngữ, mô hình này gọi là mô hình dữ liệu mức 1 (khái niệm).

- ☐ Trong phân tích thiết kế hệ thống thông tin hướng dữ liệu, CHEN- tác giả của mô hình ERD, được ANSI chọn là mô hình chuẩn cho phân tích thiết kế 1988.
- ☐ Thực thể

Ý nghĩa: Dùng biểu diễn một đối tượng trong bài toán cần phân tích

Ký hiệu:

Trong các thuộc tính Ai, có 1 (hay nhiều) thuộc tính chọn là khóa chính (primary key).

- ☐ Thuộc tính PK được gạch dưới (hoặc dùng ký hiệu qui ước khác: #)

35 csdl phân tán

Là tập hợp các csdl có quan hệ với nhau về mặt logic và được phân bố trên 1 mạng máy tính

Đặc điểm

- Điều khiển tập chung
- Độc lập dữ liệu
- Giảm dư thừa dữ liệu
- Độ tin cậy qua các giao dịch phân tán
- Cải tiến hiệu năng
- Dễ dàng mở rộng

Quy tắc phân mảnh

- Tính đầy đủ
 - Tính phục hồi
 - Tính tách biệt
-

```
USE MASTER
CREATE DATABASE QLDT
USE QLDT
CREATE TABLE SINHVIEN
(
    MSSV CHAR(8) PRIMARY KEY NOT NULL,
    TENSX NVARCHAR(30) NOT NULL,
    SODT VARCHAR(10),
    LOP CHAR(10) NOT NULL,
    DIACHI NCHAR(50) NOT NULL
)

CREATE TABLE DETAI
(
    MSDT CHAR(6) PRIMARY KEY NOT NULL,
    TENDT NVARCHAR(30) NOT NULL
)

CREATE TABLE SV_DETAI
(
    MSSV CHAR(8) FOREIGN KEY REFERENCES SINHVIEN(MSSV),
    MSDT CHAR(6) FOREIGN KEY REFERENCES DETAI(MSDT),
    CONSTRAINT PK_SV_DETAI PRIMARY KEY (MSSV,MSDT)
)

CREATE TABLE HOCHAM
(
    MSHH INT PRIMARY KEY NOT NULL,
    TENHH NVARCHAR(20) NOT NULL
)

CREATE TABLE GIAOVIEN
(
    MSGV INT PRIMARY KEY NOT NULL,
    TENGX NVARCHAR(30) NOT NULL,
    DIACHI NVARCHAR(50) NOT NULL,
    SODT VARCHAR(10) NOT NULL,
    MSHH INT FOREIGN KEY REFERENCES HOCHAM(MSHH),
    NAMHH SMALLDATETIME NOT NULL
)
```

```

CREATE TABLE HOCVI
(
    MSHV INT PRIMARY KEY NOT NULL,
    TENHV NVARCHAR(20) NOT NULL
)

CREATE TABLE CHUYENNGANH
(
    MSCN INT PRIMARY KEY NOT NULL,
    TENCN NVARCHAR(30) NOT NULL
)

CREATE TABLE GV_HV_CN
(
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV),
    MSHV INT FOREIGN KEY REFERENCES HOCVI(MSHV),
    MSCN INT FOREIGN KEY REFERENCES CHUYENNGANH(MSCN),
    NAM SMALLDATETIME NOT NULL,
    CONSTRAINT PK_GV_HV_CN PRIMARY KEY (MSGV,MSHV,MSCN)
)

CREATE TABLE GV_HDDT
(
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV),
    MSDT CHAR(6) FOREIGN KEY REFERENCES DETAI(MSDT),
    DIEM FLOAT NOT NULL,
    CONSTRAINT PK_GV_HDDT PRIMARY KEY (MSGV,MSDT)
)

CREATE TABLE GV_PBDT
(
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV),
    MSDT CHAR(6) FOREIGN KEY REFERENCES DETAI(MSDT),
    DIEM FLOAT NOT NULL,
    CONSTRAINT PK_GV_PBDT PRIMARY KEY (MSGV,MSDT)
)

CREATE TABLE GV_UVDT
(
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV),
    MSDT CHAR(6) FOREIGN KEY REFERENCES DETAI(MSDT),
    DIEM FLOAT NOT NULL,
    CONSTRAINT PK_GV_UVDT PRIMARY KEY (MSGV,MSDT)
)

```

```

CREATE TABLE HOIDONG
(
    MSHD INT PRIMARY KEY NOT NULL,
    PHONG INT,
    TGBD SMALLDATETIME,
    NGAYHD SMALLDATETIME NOT NULL,
    TINHTRANG NVARCHAR(30) NOT NULL,
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV)
)

CREATE TABLE HOIDONG_GV
(
    MSHD INT FOREIGN KEY REFERENCES HOIDONG(MSHD),
    MSGV INT FOREIGN KEY REFERENCES GIAOVIEN(MSGV),
    CONSTRAINT PK_HOIDONG_GV PRIMARY KEY (MSHD,MSGV)
)

CREATE TABLE HOIDONG_DT
(
    MSHD INT FOREIGN KEY REFERENCES HOIDONG(MSHD),
    MSDT CHAR(6) FOREIGN KEY REFERENCES DETAI(MSDT),
    QUYETDINH NCHAR(10),
    CONSTRAINT PK_HOIDONG_DT PRIMARY KEY (MSHD,MSDT)
)
/*code da dc fix, ko can chay dong nay CAI NAY THEM MOI QUNA HE*/
ALTER TABLE GIAOVIEN ADD CONSTRAINT PK_HOCHAM FOREIGN KEY (MSHH) REFERENCES
HOCHAM(MSHH);

```

A. STORED PROCEDURES VỚI THAM SỐ VÀO

1. Tham số vào là MSGV, TENGV, SODT, DIACHI, MSHH, NAMHH. Trước khi insert dữ liệu cần kiểm tra MSHH đã tồn tại trong table **HOCHAM** chưa, nếu chưa thì trả về giá trị 0.
2. Tham số vào là MSGV, TENGV, SODT, DIACHI, MSHH, NAMHH. Trước khi insert dữ liệu cần kiểm tra **MSGV** trong table GIAOVIEN có trùng không, nếu trùng thì trả về giá trị 0.
3. Giống (1) và (2) kiểm tra xem MSGV có trùng không? MSHH có tồn tại chưa? Nếu MSGV trùng thì trả về 0. Nếu MSHH chưa tồn tại trả về 1, ngược lại cho insert dữ liệu.
4. Đưa vào MSDT cũ, TENDT mới. Hãy cập nhật tên đề tài mới với mã đề tài cũ không đổi nếu không tìm thấy trả về 0, ngược lại cập nhật và trả về 1.

5. Tham số đưa vào MSSV, TENSX mới, DIACHI mới thủ lục dùng để cập nhật sinh viên trên, nếu không tìm thấy trả về 0, ngược lại cập nhật và trả về 1.

B. STORED PROCEDURES VỚI THAM SỐ VÀO VÀ RA

1. Đưa vào TENHV trả ra: Số GV thỏa học vị, nếu không tìm thấy trả về 0.

2. Đưa vào MSDT cho biết: Điểm trung bình của đề tài, nếu không tìm thấy trả về 0.

3. Đưa vào TENGX trả ra: SDT của giáo viên đó, nếu không tìm thấy trả về 0. Nếu trùng tên thì có báo lỗi không? Tại sao? Làm sao để hiện thông báo có bao nhiêu giáo viên trùng tên và trả về các SDT.

4. Đưa vào MSHD trả ra: Tên chủ tịch hội đồng và SDT, nếu không tìm thấy trả về 0.

5. Đưa vào TENGX cho biết: Số đề tài hướng dẫn, số đề tài phản biện do giáo viên đó phụ trách. Nếu trùng tên thì có báo lỗi không hay hệ thống sẽ đếm tất cả các đề tài của những giáo viên trùng tên đó? Tại sao? Làm sao để hiện thông báo có bao nhiêu giáo viên trùng tên và trả về thông tin được yêu cầu. Cần lưu ý gì với tham số vào không để không xảy ra lỗi tương tự hoặc tính hết các trường hợp để không báo lỗi và kết quả trả về đúng?

6. Đưa vào MSHD cho biết: Điểm trung bình các đề tài của hội đồng đó.

C. TRIGGER

1. Tạo Trigger thỏa mãn điều kiện khi xóa một đề tài sẽ xóa các thông tin liên quan.

2. Tạo Trigger thỏa mãn ràng buộc là khi đổi 1 mã số giáo viên (MSGV) thì sẽ thay đổi các thông tin liên quan.

3. Tạo Trigger thỏa mãn ràng buộc là một hội đồng không quá 10 đề tài. Dùng “Group by” có được không? Giải thích.

4. Tạo Trigger thỏa mãn ràng buộc là một đề tài không quá 3 sinh viên. Dùng “Group by” có được không? Giải thích.

5. Tạo Trigger thỏa mãn ràng buộc là một giáo viên muốn có học hàm PGS phải là tiến sĩ.

--PHẦN BT A

--CÂU:1

CREATE PROC A_1

 @MSGV INT,@TENG NVARCHAR(30),@DIACHI NVARCHAR(50),@SDT VARCHAR(10),@MSHH
INT,@NAMHH SMALLDATETIME

AS

BEGIN

 IF EXISTS (SELECT MSHH FROM HOCHAM WHERE MSHH=@MSHH)

 BEGIN

 INSERT INTO GIAOVIEN VALUES

(@MSGV,@TENG NVARCHAR(30),@DIACHI NVARCHAR(50),@SDT VARCHAR(10),@MSHH,@NAMHH)

 PRINT 'DA CHEN THANH CONG'

 END

 ELSE

 BEGIN

 PRINT 'MSHH KHONG TON TAI'

 RETURN 0

 END

END

--CHAY PROC

EXEC A_1 112,TEST1,HCM,11112222,2,1996--DA CHAY DUOC

--KIEM TRA DU LIEU DA DUOC DUA VAO CHUA

SELECT * FROM GIAOVIEN

--XOA DU LIEU VUA DUA VAO

DELETE FROM GIAOVIEN WHERE MSGV=112--DA XOA DU LIEU THANH CONG

--CÂU:2

CREATE PROC A_2

 @MSGV INT,@TENG NVARCHAR(30),@DIACHI NVARCHAR(50),@SDT VARCHAR(10),@MSHH
INT,@NAMHH SMALLDATETIME

AS

BEGIN

 IF EXISTS (SELECT MSGV FROM GIAOVIEN WHERE MSGV=@MSGV)

 BEGIN

```

            INSERT INTO GIAOVIEN VALUES
(@MSGV,@TENGV,@DIACHI,@SDT,@MSHH,@NAMHH)
            PRINT 'DA CHEN THANH CONG'
        END
    ELSE
        BEGIN
            PRINT 'MSGV KHONG TON TAI'
            RETURN 0
        END
    END
END

--CÂU:3
CREATE PROC A_3
    @MSGV INT,@TENGV NVARCHAR(30),@DIACHI NVARCHAR(50),@SDT VARCHAR(10),@MSHH
INT,@NAMHH SMALLDATETIME
AS
BEGIN
    IF EXISTS (SELECT MSGV FROM GIAOVIEN WHERE MSGV=@MSGV)
        BEGIN
            PRINT 'CHEN THAT BAI, MSGV BI TRUNG'
            RETURN 0
        END
    ELSE
        BEGIN
            IF EXISTS (SELECT MSHH FROM GIAOVIEN WHERE MSHH=@MSHH)
                BEGIN
                    INSERT INTO GIAOVIEN VALUES
(@MSGV,@TENGV,@DIACHI,@SDT,@MSHH,@NAMHH)
                    PRINT 'DA CHEN THANH CONG'
                END
            ELSE
                BEGIN
                    PRINT 'CHEN THAT BAI, MSHH CHUA TON TAI'
                    RETURN 1
                END
            END
        END
    END
END

--CÂU:4
CREATE PROC A_4
    @MSDT CHAR(6),@TENDT NVARCHAR(30)

```

```

AS
BEGIN
    IF EXISTS (SELECT MSDT FROM DETAI WHERE MSDT=@MSDT)
        BEGIN
            UPDATE DETAI SET TENDT = @TENDT WHERE MSDT = (SELECT MSDT FROM
DETAI WHERE MSDT=@MSDT)
            PRINT 'DA CAP NHAP THANH CONG TENDT'
            RETURN 1
        END
    ELSE
        BEGIN
            PRINT 'KHONG TON TAI MSDT'
            RETURN 0;
        END
    END
--CÂU:5
CREATE PROC A_5
    @MSSV CHAR(8), @TENS NVARCHAR(30),@DIACHI NCHAR(50)
AS
BEGIN
    IF EXISTS (SELECT MSSV FROM SINHVIEN WHERE MSSV=@MSSV)
        BEGIN
            UPDATE SINHVIEN SET TENS = @TENS, DIACHI = @DIACHI WHERE MSSV =
@MSSV
            PRINT 'CAP NHAP SINH VIEN THANH CONG'
            RETURN 1
        END
    ELSE
        BEGIN
            PRINT 'CAP NHAP KHONG THANH CONG'
            RETURN 0
        END
    END
--PHẦN BT B
--CÂU:1
CREATE PROC B_1
    @TEN NVARCHAR(20), @SOGV INT OUTPUT
AS
BEGIN

```

```

IF EXISTS (SELECT TENHV FROM HOCVI WHERE TENHV=@TENHV)
    BEGIN
        SELECT @SOGV = COUNT (*) FROM GIAOVIEN T1, HOCVI T2, GV_HV_CN T3
        WHERE T1.MSGV=T3.MSGV AND T2.MSHV=T3.MSHV AND T2.TENHV = @TENHV
        PRINT 'VỚI HỌC VỊ '+@TENHV + ' CÓ ' + CAST(@SOGV AS VARCHAR)
    END
ELSE
    BEGIN
        PRINT 'VỚI HỌC VỊ '+@TENHV + ' KHÔNG CÓ GV NÀO'
        RETURN 0
    END
END

--THUC THI,TEST CAU LENH--*checked
DECLARE @TENHV1 NVARCHAR(20), @SOGV1 INT
SET @TENHV1 = N'Kỹ sư'
EXEC B_1 @TENHV1,@SOGV1 OUTPUT

--CÂU:2
CREATE PROC B_2
    @MSDT CHAR(6), @DIEMTB FLOAT OUTPUT
AS
BEGIN
    IF EXISTS (SELECT MSDT FROM DETAI WHERE MSDT = @MSDT)
        BEGIN
            SET @DIEMTB = ((SELECT AVG(DIEM) FROM GV_UVDT WHERE MSDT=@MSDT)
+ (SELECT AVG(DIEM) FROM GV_PBDT WHERE MSDT=@MSDT) + (SELECT AVG(DIEM) FROM GV_HDDT
WHERE MSDT=@MSDT))/3
            PRINT 'VOI MA SO DE TAI ' + @MSDT + ' CO ' + CAST(@DIEMTB AS VARCHAR)
        END
    ELSE
        BEGIN
            PRINT 'KHONG TIM THAY MSDT'
            RETURN 0
        END
    END
END

--THUC THI, TEST THU, *CHECKED
DECLARE @MSDT1 CHAR(6), @DIEMTB1 FLOAT
SET @MSDT1 = '97002'

```

```
EXEC B_2 @MSDT1,@DIEMTB1 OUTPUT
```

```
/* TEST THU CHO CAU 2B
```

```
DECLARE @S FLOAT
```

```
SET @S=((SELECT AVG(DIEM) FROM GV_UVDT WHERE MSDT='97005') + (SELECT AVG(DIEM) FROM  
GV_UVDT WHERE MSDT='97003'))/2
```

```
PRINT @S
```

```
SELECT AVG(DIEM) FROM GV_UVDT WHERE MSDT='97005'
```

```
*/
```

```
--CÂU:3
```

```
CREATE PROC B_3
```

```
    @TENGV NVARCHAR(30), @SDT VARCHAR(10) OUTPUT
```

```
AS
```

```
BEGIN
```

```
    IF EXISTS (SELECT TENGV FROM GIAOVIEN WHERE TENGV = @TENGV)
```

```
        BEGIN
```

```
            SET @SDT = (SELECT SODT FROM GIAOVIEN WHERE TENGV = @TENGV)
```

```
            PRINT 'SDT CUA GV ' + @TENGV + ' LA ' + @SDT
```

```
        END
```

```
    ELSE
```

```
        BEGIN
```

```
            RETURN 0
```

```
        END
```

```
END
```

--Khi gặp phải tên giáo viên trùng nhau sẽ không bị báo lỗi, thay vào đó sẽ lấy tên giáo viên đầu tiên tìm được

--Để hiển thị số giáo viên có tên trùng nhau ta sử dụng câu truy vấn sau

```
SELECT COUNT(*) FROM GIAOVIEN WHERE TENGV = @TENGV
```

--Để hiển thị tất cả số đt của các giáo viên trùng tên ta dùng câu truy vấn sau

```
SELECT @SDT FROM GIAOVIEN WHERE MSGV IN (SELECT MSGV FROM GIAOVIEN WHERE  
TENGV=@TENGV)
```

--Hiện thông báo có bao nhiêu giáo viên trùng tên và trả về các SDT.

--Ta sẽ viết lại PROC NHU SAU

```
CREATE PROC B_3_FIX
```

```
    @TENGV NVARCHAR(30), @SDT VARCHAR(10) OUTPUT
```

```
AS
```

```
BEGIN
```

```
    IF EXISTS (SELECT TENGV FROM GIAOVIEN WHERE TENGV = @TENGV)
```

```
        BEGIN
```

```

        SET @SDT = (SELECT SODT FROM GIAOVIEN WHERE MSGV IN (SELECT MSGV
FROM GIAOVIEN WHERE TENGV=@TENGV))
        PRINT 'GIAO VIEN TEN ' + @TENGV + ' SO NGUOI TRUNG TEN LA ' + (SELECT
COUNT(*) FROM GIAOVIEN WHERE TENGV = @TENGV) + ' SDT LAN LUOT CUA HO LA: ' + @SDT
    END
ELSE
    BEGIN
        RETURN 0
    END
END

```

--CÂU:4

```

CREATE PROC B_4
    @MSHD INT,@TENGV NVARCHAR(30) OUTPUT,@SDT VARCHAR(10) OUTPUT
AS
BEGIN
    IF EXISTS (SELECT MSHD FROM HOIDONG WHERE MSHD=@MSHD)
        BEGIN
            SET @TENGV = (SELECT TENGV FROM GIAOVIEN GV, HOIDONG HD WHERE
GV.MSGV=HD.MSGV AND HD.MSHD=@MSHD)
            SET @SDT = (SELECT SODT FROM GIAOVIEN GV, HOIDONG HD WHERE
GV.MSGV=HD.MSGV AND HD.MSHD=@MSHD)
            PRINT 'VOI MSHD LA: ' + @MSHD + ' TEN CHU TICH HD LA: ' + @TENGV + ' SDT
CUA CHU TICH: ' + @SDT
        END
    ELSE
        BEGIN
            RETURN 0
        END
    END
END

```

--CÂU:5

```

CREATE PROC B_5
    @TENGV VARCHAR(30), @SODTHD INT OUTPUT, @SODTPT INT OUTPUT
AS
BEGIN
    IF EXISTS (SELECT MSGV FROM GIAOVIEN WHERE TENGV = @TENGV)
        BEGIN
            DECLARE @SOGV INT
            SELECT @SOGV = COUNT(*) FROM GIAOVIEN WHERE TENGV = @TENGV
            IF (@SOGV = 1)

```

```

        BEGIN
            SELECT @SODTHD = COUNT(*) FROM GV_HDDT A1,GIAOVIEN
A2 WHERE A1.MSGV=A2.MSGV AND A2.TENGV = @TENGV
            SELECT @SODTPT = COUNT(*) FROM GV_PBDT A1,GIAOVIEN A2
WHERE A1.MSGV=A2.MSGV AND A2.TENGV = @TENGV
        END
    ELSE
        BEGIN
            IF (@SOGV>1)
                BEGIN
                    PRINT 'CO ' + CAST(@SOGV AS VARCHAR) + 'GV
TRUNG TEN'
                END
            ELSE
                BEGIN
                    PRINT'KHONG CO GIAO VIEN NAO CO TEN
'+@TENGV
                END
            RETURN 0
        END
    END
END
--CÂU:6
CREATE PROC B_6
    @MSHD INT,@DIEM_HDTB FLOAT OUTPUT
AS
BEGIN
    IF EXISTS (SELECT MSHD FROM HOIDONG_DT WHERE MSHD=@MSHD)
        BEGIN
            --TÍNH ĐIỂM TB CỦA MỖI HỘI ĐỒNG TRÊN MỖI BẢNG RỒI CỘNG LẠI CHIA CHO
SỐ BẢNG--XEM COMMENT BÊN DƯỚI
            SET @DIEM_HDTB = ((SELECT AVG(DIEM) FROM GV_HDDT WHERE MSMT IN
(SELECT MSMT FROM HOIDONG_DT WHERE MSHD=@MSHD)) +(SELECT AVG(DIEM) FROM GV_PBDT
WHERE MSMT IN (SELECT MSMT FROM HOIDONG_DT WHERE MSHD=@MSHD)) +(SELECT AVG(DIEM)
FROM GV_UVDT WHERE MSMT IN (SELECT MSMT FROM HOIDONG_DT WHERE MSHD=@MSHD)))/3
            PRINT 'DIEM TB CUA HOI DONG SO '+ @MSHD + ' LA: ' +@DIEM_HDTB
        END
    ELSE
        BEGIN
            PRINT 'MSHD KHONG HOP LE'
        END
    END
END

```

```

END

END

/*GIẢI THÍCH CÂU 6
TÍNH ĐIỂM TB CỦA MỖI HỘI ĐỒNG TRÊN MỖI BẢNG RỒI CỘNG LẠI CHIA CHO SỐ BẢNG
-----@B1_HDDT FLOAT,@B2_PBDT FLOAT,@B3_UVDT
FLOAT
    SET @B1_HDDT= (SELECT AVG(DIEM) FROM GV_HDDT WHERE MSDT IN (SELECT MSDT FROM
HOIDONG_DT WHERE MSHD=@MSHD))
    SET @B2_PBDT= (SELECT AVG(DIEM) FROM GV_PBDT WHERE MSDT IN (SELECT MSDT FROM
HOIDONG_DT WHERE MSHD=@MSHD))
    SET @B3_UVDT= (SELECT AVG(DIEM) FROM GV_UVDT WHERE MSDT IN (SELECT MSDT FROM
HOIDONG_DT WHERE MSHD=@MSHD))
    SET @DIEM_HDTB = (@B1_HDDT +@B2_PBDT +@B3_UVDT )/3
    PRINT 'DIEM TB CUA HOI DONG SO '+' @MSHD + ' LA: ' +@DIEM_HDTB
*/
/* CÂU LỆNH TEST KIỂM TRA CHO MỘT PHẦN CÂU 6 *CHECKED
--CÂU LỆNH NÀY CÓ NGHĨA, TÍNH ĐIỂM TRUNG BÌNH MỘT HỘI ĐỒNG KHI ĐƯA MSHD VÀO, CỦA BẢNG
GV_HDDT
DECLARE @DIEM_HDTB FLOAT
SET @DIEM_HDTB = (SELECT AVG(DIEM) FROM GV_HDDT WHERE MSDT IN (SELECT MSDT FROM
HOIDONG_DT WHERE MSHD=3))
PRINT @DIEM_HDTB
*/

--PHẦN BT C
--CÂU 1:
CREATE TRIGGER C_1 ON DETAI
FOR DELETE
AS
BEGIN
    IF(@@ROWCOUNT=0)
        BEGIN
            PRINT'TABLE DETAI KHONG CO DU LIEU'
            RETURN
        END
    DELETE SV_DETAI FROM SV_DETAI T1, deleted T2 WHERE T1.MSDT=T2.MSDT
    DELETE GV_HDDT FROM GV_HDDT T1, deleted T2 WHERE T1.MSDT=T2.MSDT

```



```

DELETE GV_PBDT FROM GV_PBDT T1, deleted T2 WHERE T1.MSDT=T2.MSDT
DELETE GV_UVDT FROM GV_UVDT T1, deleted T2 WHERE T1.MSDT=T2.MSDT
DELETE HOIDONG_DT FROM HOIDONG_DT T1,deleted T2 WHERE T1.MSDT=T2.MSDT
PRINT'XOA THANH CONG'

END
--TẠO THỦ TỤC KIỂM TRA TRIGGER
CREATE PROC KT_C_1
    @MSDT CHAR(6)
AS
BEGIN
    IF EXISTS (SELECT * FROM DETAI WHERE MSDT=@MSDT)
        BEGIN
            --KHÓA DÀNG BƯỚC
            ALTER TABLE SV_DETAI NOCHECK CONSTRAINT ALL
            ALTER TABLE GV_HDDT NOCHECK CONSTRAINT ALL
            ALTER TABLE GV_PBDT NOCHECK CONSTRAINT ALL
            ALTER TABLE GV_UVDT NOCHECK CONSTRAINT ALL
            ALTER TABLE HOIDONG_DT NOCHECK CONSTRAINT ALL
            --TRIGGER THỰC THI
            DELETE FROM DETAI WHERE MSDT=@MSDT
            --KÍCH HOẠT LẠI
            ALTER TABLE SV_DETAI CHECK CONSTRAINT ALL
            ALTER TABLE GV_HDDT CHECK CONSTRAINT ALL
            ALTER TABLE GV_PBDT CHECK CONSTRAINT ALL
            ALTER TABLE GV_UVDT CHECK CONSTRAINT ALL
            ALTER TABLE HOIDONG_DT CHECK CONSTRAINT ALL
        END
    ELSE
        PRINT'KHONG CO DE TAI: '+@MSDT
END

--THỰC THI
EXEC KT_C_1 '1'
SELECT * FROM SV_DETAI

--CÂU 2:
CREATE TRIGGER C_2 ON GIAOVIEN FOR UPDATE
AS
BEGIN
    IF(@@ROWCOUNT = 0)

```

```

BEGIN
    PRINT'TABLE KHONG CO GI HET'
    RETURN
END
IF UPDATE(MSGV)
BEGIN
    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM GV_HV_CN A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM GV_HDDT A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM GV_PBDT A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM GV_UVDT A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM HOIDONG A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    UPDATE A1 SET A1.MSGV = A3.MSGV
    FROM HOIDONG A1, deleted A2, inserted A3
    WHERE A1.MSGV = A2.MSGV

    PRINT'CAP NHAP THANH CONG'
END
END

--CÂU 3
CREATE TRIGGER C_3 ON HOIDONG_DT FOR INSERT, UPDATE
AS
BEGIN
    IF(SELECT COUNT(A1.MSHD) FROM HOIDONG_DT A1,inserted A2 WHERE
    A1.MSHD=A2.MSHD)>=10

```

```

BEGIN
    PRINT'KHONG THE QUA 10 DE TAI'
    ROLLBACK TRAN
END
END

```

--CÂU 4:

```

CREATE TRIGGER C_4 ON SV_DETAI FOR INSERT,UPDATE
AS
BEGIN
    IF((SELECT COUNT(A1.MSSV) FROM SV_DETAI A1, inserted A2 WHERE A1.MSDT=A2.MSDT)<=3)
    BEGIN
        DECLARE @TENDT VARCHAR(30), @SOSV INT
        SELECT @TENDT = A1.TENDT FROM DETAI A1,SV_DETAI A2,inserted A3 WHERE
A1.MSDT=A2.MSDT AND A2.MSDT = A3.MSDT
        SELECT @SOSV = COUNT(A1.MSSV) FROM SV_DETAI A1,inserted A2 WHERE
A1.MSDT=A2.MSDT
        PRINT'DETAI '+@TENDT + ' CO ' + CAST(@SOSV AS VARCHAR) + ' SV'
    END
    ELSE
    BEGIN
        PRINT'KHONG DUOC QUA 3 SV 1 DE TAI'
        ROLLBACK TRAN
    END
END
END

```

--CÂU 5:

```

CREATE TRIGGER C_5 ON GIAOVIEN FOR INSERT,UPDATE
AS
BEGIN
    IF UPDATE(MSHH)
    BEGIN
        IF EXISTS (SELECT A1.MSHH FROM inserted A1,HOCHAM A2 WHERE A1.MSHH =
A2.MSHH AND A2.TENHH LIKE '%PHÓ GIÁO SƯ%')
        BEGIN
            IF NOT EXISTS (SELECT A1.MSHV FROM GV_HV_CN A1,HOCVI A2,inserted A3
WHERE A2.MSHV=A1.MSHV AND A1.MSGV = A3.MSGV AND A2.TENHV = 'Tiến sĩ')
            BEGIN

```

```

                PRINT'CHUA THE LAM PGS VI CHUA DUOC LAM TIEN SI'
                ROLLBACK TRAN
            END
        END
    END
ELSE
    BEGIN
        IF EXISTS (SELECT A1.MSHH FROM inserted A1,HOCHAM A2 WHERE A1.MSHH =
A2.MSHH AND A2.TENHH NOT LIKE '%PHÓ GIÁO SƯ%')
            BEGIN
                PRINT'KHONG CO HOC HAM PGS VI CHUA CO HOC VI TIEN SI'
                ROLLBACK TRAN
            END
        END
    END
END

```