MỤC LỤC

LỜI CẢM ƠN

Trước hết em xin gửi lời cảm ơn chân thành tới cô giáo ThS.Nguyễn Thị Xuân Hương và KS. Đào Quang Huynh đã tận tình chỉ bảo và hướng dẫn em hoàn thành tốt đề tài tốt nghiệp này.

Em xin chân thành cảm ơn các thầy cô giáo ở khoa Công nghệ thông tin trường Đại Học Dân Lập Hải Phòng đã giảng dạy và chỉ bảo cho em trong 1,5 năm học tại trường, để em có được các kiến thức cơ bản phục vụ cho quá trình làm tốt nghiệp.

Cuối cùng em xin bày tỏ lòng biết ơn tới những người thân trong gia đình và các bạn bè đã chia sẻ và động viên em trong suốt quá trình học tập cho đến nay.

Hải Phòng, ngày tháng năm 2009
Sinh viên

Phạm Minh Tiến

LỜI NÓI ĐẦU

Từ vài thập niên trở lại đây, với những tác động mạnh mẽ của các tiến bộ trong công nghệ phần cứng và truyền thông, các hệ thống dữ liệu phục vụ cho các lĩnh vực kinh tế - xã hội đã phát triển bùng nổ, lượng dữ liệu được tạo ra ngày càng lớn. Sự phong phú về dữ liệu, thông tin cùng với khả năng kịp thời khai thác chúng đã mang đến những năng suất và chất lượng mới cho công tác quản lý, hoạt động kinh doanh,…Nhưng rồi các yêu cầu về thông tin trong các lĩnh vực hoạt động đó, đặc biệt trong lĩnh vực làm ra quyết định, ngày càng đòi hỏi cao hơn, người quyết định không những cần dữ liệu mà còn cần có thêm nhiều hiểu biết, nhiều tri thức để hỗ trợ cho việc ra quyết định của mình. Cho đến những năm 90 của thế kỷ trước, nhu cầu khám phá tri thức mới thực sự bùng nổ, theo đó, hàng loạt các lĩnh vực nghiên cứu về tổ chức các kho dữ liệu và kho thông tin, các hệ trợ giúp quyết định, các thuật toán nhận dạng mẫu và phân lớp mẫu, … ra đời, một trong số đó là phân cụm dữ liệu (Data Clustering). Phân cụm dữ liệu là quá trình tìm kiếm và phát hiện ra các cụm hoặc các mẫu dữ liệu tự nhiên trong cơ sở dữ liệu lớn. Các kỹ thuật chính được áp dụng trong phân cụm dữ liệu phần lớn được kế thừa từ lĩnh vực thống kê, học máy, nhận dạng, lượng hoá,.. Đến nay, đã có nhiều ứng dụng phân cụm dữ liệu cho việc giải quyết các vấn đề trong các lĩnh vực như tài chính, thông tin địa lý, sinh học, nhận dạng ảnh, …Trong thời gian gần đây, trong lĩnh vực PCDL, người ta tập trung chủ yếu vào nghiên cứu, phân tích các mô hình dữ liệu phức tạp như dữ liệu văn bản, Web, hình ảnh.....

Hiện nay, Oracle là một hệ quản trị CSDL đang được sử dụng rộng rãi, đặc biệt là trong các cơ quan, tổ chức có nhu cầu lưu trữ một lượng dữ liệu lớn. Tuy nhiên, với khối dữ liệu khổng lồ như vậy, việc khai thác hữu ích các thông tin trong đó là một yêu cầu rất cáp thiết. Từ phiên bản Oracle9i đã tích hợp kỹ thuật khai phá dữ liệu trong phiên bản này để trợ giúp cho người sử dụng có thể tìm kiếm các thông tin cần khai thác. Vì vậy, em chọn đề tài " Tìm hiểu về kĩ thuật phân cụm dữ liệu trong hệ quản trị cơ sở dữ liệu Oracle "làm đề tài tốt nghiệp cho mình với mục đích là vận dụng các kiến thức đã học và nghiên cứu các vấn đề mới để xây dựng một ứng dụng trong hệ quản trị CSDL Oracle có áp dụng kỹ thuật phân cụm.

Nội dung của đồ án gồm 4 chương:

Chương 1: Phân cụm dữ liệu : trong chương này em trình bày tổng quan về phân cụm dữ liệu bao gồm các kiểu dữ liệu có thể phân cụm , các ứng dụng và các kỹ thuật phân cụm dữ liệu .

Chương 2: Hệ quản trị cơ sở dữ liệu Oracle

Giới thiệu về hệ quản trị cơ sở dữ liệu Oracle và phân cụm dữ liệu trong Oracle

Chương 3: Mô hình Use Case

Giới thiệu mô hình Use Case , biểu đồ và quan hệ use case .

Chương 4: Chương trình ứng dụng:

Giới thiệu chương trình ứng dụng quản lý văn bản đến và đi , sử dụng mô hình Use case , cơ sở dữ liệu Oracle có sử dụng kĩ thuật phân cụm dữ liệu để phân cụm văn bản đến và đi trong Oracle

Phần kết luận trình bày tóm tắt các kết quả thu được và các đề xuất cho hướng phát triển của đề tài .

Chương 1 PHÂN CỤM DỮ LIỆU

1.1 Kỹ thuật phân cụm dữ liệu.

PCDL là một kĩ thuật trong Data Mining (khai phá dữ liệu), nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tự nhiên tiềm ẩn cần quan tâm trong tập dữ liệu lớn, từ đó cung cấp thông tin, tri thức hữu ích cho ra quyết định.

Mục tiêu chính của phương pháp phân cụm dữ liệu là nhóm các đối tượng tương tự nhau trong tập dữ liệu vào các *cụm* sao cho các đối tượng thuộc cùng một lớp là "tương đồng"còn các đối tượng thuộc các cụm khác nhau sẽ "không tương đồng". Phân cụm dữ liệu được sử dụng nhiều trong các ứng dụng về phân loại văn bản, phân đoạn khách hàng, nhận dạng mẫu, phân loại trang Web…

1.2 Các ứng dụng của phân cụm dữ liệu

Một số ứng dụng điển hình phân cụm dữ liệu trong các lĩnh vực sau:

- *Thương mại*: Trong thương mại, PCDL có thể giúp các thương nhân khám phá ra các nhóm khách hàng quan trọng có các đặc trưng tương đồng nhau và đặc tả họ từ các mẫu mua bán trong CSDL khách hàng.
- *Sinh học*: Trong sinh học, PCDL được sử dụng để xác định các loại sinh vật, phân loại các Gen với chức năng tương đồng và thu được các cấu trúc trong các mẫu.
- *Phân tích dữ liệu không gian*: PCDL có thể trợ giúp người dùng tự động phân tích và xử lý các dữ liêu không gian như nhận dạng và chiết xuất các đặc tính hoặc các mẫu dữ liệu quan tâm có thể tồn tại trong CSDL không gian.
- *Lập quy hoạch đô thị:* Nhận dạng các nhóm nhà theo kiểu và vị trí địa lý,…nhằm cung cấp thông tin cho quy hoạch đô thị.
- *Nghiên cứu trái đất*: Phân cụm để theo dõi các tâm động đất nhằm cung cấp thông tin cho nhận dạng các vùng nguy hiểm.
- *Địa lý:* Phân lớp các động vật và thực vật và đưa ra đặc trưng của chúng.
- *Web Mining*: PCDL có thể khám phá các nhóm tài liệu quan trọng, có nhiều ý nghĩa trong môi trường Web. Các lớp tài liệu này trợ giúp cho việc khám phá tri thức từ dữ liệu,…

1.3 Các kiểu dữ liệu và độ đo tương tự

Phân cụm dữ liệu là quá trình phân chia một tập dữ liệu ban đầu thành các cụm sao cho các đối tượng trong cùng một cụm "tương tự". Việc tính "khoảng cách" giữa các đối tượng, hay phép đo tương tự giữa các cặp đối tượng để phân chia chúng vào các cụm khác nhau. Dựa vào hàm tính độ tương tự này cho phép xác định được hai đối tượng có tương tự hay không. Theo quy ước, giá trị của hàm tính độ đo tương tự càng lớn thì sự tương đòng giữa các đối tượng càng lớn và ngược lại. Hàm tính độ phi tương tự tỉ lệ nghịch với hàm tính độ tương tự.

Các kiểu dữ liệu thường được sử dụng trong PCDL. Trong PCDL, các đối tượng dữ liệu cần phân tích có thể là *con người, cái nhà, tiền lương, các thực thể phần mềm,....* Các đối tượng này thường được diễn tả dưới dạng các thuộc tính của nó

Có 2 cách phân loại các kiểu thuộc tính: Dựa trên kích thước miền (Domain size) & Dựa trên hệ đo (Measurement Scale).

1.3.1 Phân loại các kiểu dữ liệu dựa trên kích thước miền

- *Thuộc tính liên tục (Continuous Attribute)*: nghĩa là giữa hai giá trị tồn tại vô số giá trị khác. Thí dụ như các thuộc tính về màu, nhiệt độ hoặc cường độ âm thanh.
- *Thuộc tính rời rạc (DiscretteAttribute)*: Nếu miền giá trị của nó là tập hữu hạn, đếm được. Thí dụ như các thuộc tính về *số serial của một cuốn sách, số thành viên trong một gia đình*, …

Lớp các thuộc tính nhị phân là trường hợp đặc biệt của thuộc tính rời rạc mà miền giá trị của nó chỉ có 2 phần tử được diễn tả như: *Yes / No* hoặc *Nam/Nữ*, *False/true*,…

1.3.2 Phân loại các kiểu dữ liệu dựa trên hệ đo

Giả sử có hai đối tượng x, y và các thuộc tính x_i, y_i tương ứng với thuộc tính thứ i của chúng. Chúng ta có các lớp kiểu dữ liệu như sau:

- *Thuộc tính định danh (nominal Scale, tên)*: nếu x và y là hai đối tượng thuộc tính thì chỉ có thể xác định là $x \neq y$ hoặc x=y. Thí dụ như thuộc tính về *nơi sinh* hoặc thuộc tính *các đội bóng chơi cho giải vô địch quốc gia Việt Nam*.
- *Thuộc tính có thứ tự (Ordinal Scale)*: là thuộc tính định danh có thêm tính *thứ tự,* nhưng chúng không được định lượng. Nếu x và y là hai thuộc tính

thứ tự thì ta có thể xác định là $x \neq y$ hoặc x=y hoặc x>y hoặc x<y. Thí dụ như thuộc tính *Huy chương* của vận động viên thể thao.

- *Thuộc tính khoảng (Interval Scale)*: Với thuộc tính khoảng, chúng ta có thể xác định một thuộc tính là đứng trước hoặc đứng sau thuộc tính khác với một khoảng là bao nhiêu. Nếu $x_i > y_i$ thì ta nói x cách y một khoảng $x_i - y_i$ tương ứng với thuộc tính thứ i. Thí dụ về thuộc tính khoảng như thuộc tính *số kênh* trên truyền hình.
- *Thuộc tính tỉ lệ (Ratio Scale):* là thuộc tính khoảng nhưng được xác định một cách tương đối so với điểm mốc đầy ý nghĩa, *thí dụ như thuộc tính chiều cao hoặc cân nặng lấy điểm 0 làm mốc.*

Chó ý:

- Thuộc tính định danh và thuộc tính có thứ tự gọi chung là thuộc tính hạng mục
- Thuộc tính khoảng và thuộc tính tỉ lệ được gọi là thuộc tính số.

1.4 Một số kỹ thuật tiếp cận trong phân cụm dữ liệu

Các kỹ thuật áp dụng để giải quyết vấn đề phân cụm dữ liệu đều hướng tới 2 mục tiêu chung: *Chất lượng của các cụm khám phá được và tốc độ thực hiện của thuật toán.* Hiện nay, các kỹ phân cụm dữ liệu có thể phân loại theo các cách tiếp cận chính sau.

1.4.1 Phân cụm phân hoạch

Ta phân 1 tập dữ liệu có n phần tử cho trước thành k nhóm dữ liệu sao cho: mỗi phần tử dữ liệu chỉ thuộc về 1 nhóm dữ liệu và mỗi nhóm dữ liệu có tối thiểu ít nhất 1 phần tử dữ liệu.

Một số thuật toán phân cụm phân hoạch điển hình như k-means, PAM, CLARA, CLARANS,…

1.4.2 Phân cụm dữ liệu phân cấp

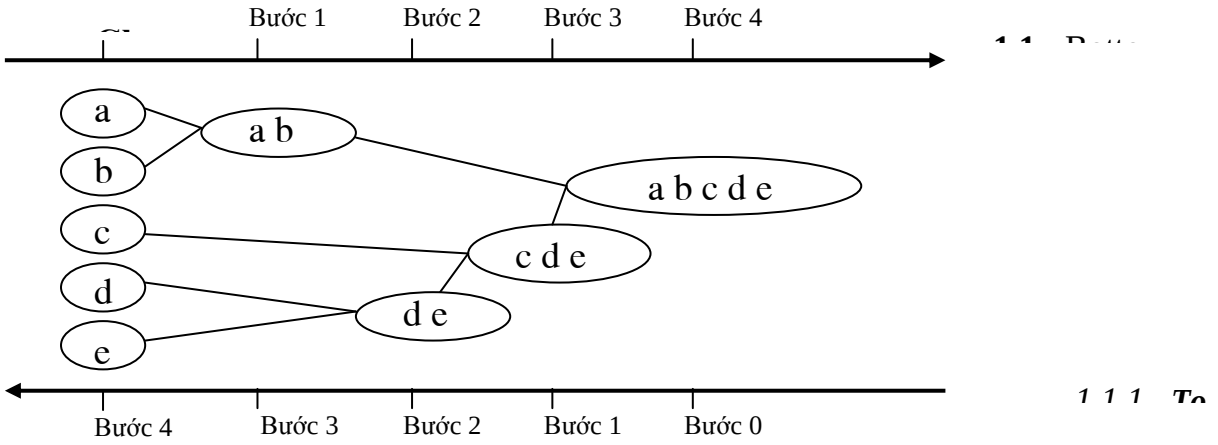
Phân cụm phân cấp sắp xếp một tập dữ liệu đã cho thành một cấu trúc có dạng hình cây, cây phân cấp này được xây dựng theo kỹ thuật đệ quy. Cây phân cụm có thể được xây dựng theo hai phương pháp tổng quát:

- **Phương pháp "dưới lên" (***Bottom up***)**: Phương pháp này bắt đầu với mỗi đối tượng được khởi tạo tương ứng với các cụm riêng biệt, sau đó tiến hành nhóm

các đối tượng theo một độ đo tương tự (như khoảng cách giữa hai trung tâm của hai nhóm), quá trình này được thực hiện cho đến khi tất cả các nhóm được hòa nhập vào một nhóm (mức cao nhất của cây phân cấp) hoặc cho đến khi các điều kiện kết thúc thỏa mãn. Như vậy, cách tiếp cận này sử dụng chiến lược ăn tham trong quá trình phân cụm.

- **Phương pháp "trên xuống" (*Top Down*)**: Bắt đầu với trạng thái là tất cả các đối tượng được xếp trong cùng một cụm. Mỗi vòng lặp thành công, một cụm được tách thành các cụm nhỏ hơn theo giá trị của một phép đo độ tương tự nào đó cho đến khi mỗi đối tượng là một cụm, hoặc cho đến khi điều kiện dừng thỏa mãn. Cách tiếp cận này sử dụng chiến lược chia để trị trong quá trình phân cụm.

Thí dụ: Hình 4 dưới đây là một thí dụ sử dụng hai chiến lược phân cụm phân cấp khác nhau như đã trình bày ở trên.



Hình 4: Các chiến lược phân cụm phân cấp

Một số thuật toán phân cụm phân cấp điển hình như CURE, BIRCH, …

1.4.3 Phân cụm dữ liệu dựa trên mật độ

Phương pháp này nhóm các đối tượng theo hàm mật độ xác định. Mật độ được định nghĩa như là số các đối tượng lân cận của 1 đối tượng dữ liệu theo một ngưỡng nào đó.

Một số thuật toán PCDL dựa trên mật độ điển hình như DBSCAN, OPTICS, DENCLUE, …

1.4.4 Phân cụm dữ liệu dựa trên lưới

Phương pháp này chủ yếu tập trung áp dụng cho lớp dữ liệu không gian.

Một số thuật toán PCDL dựa trên cấu trúc lưới điển hình như: STING, WAVECluster, CLIQUE,…

1.4.5 Phân cụm dữ liệu dựa trên mô hình

Có hai tiếp cận chính: *Mô hình thống kê và Mạng Nơ ron*

1.4.6 Phân cụm dữ liệu có ràng buộc

Để phân cụm dữ liệu không gian hiệu quả hơn, các nghiên cứu bổ sung cần được thực hiện để cung cấp cho người dùng khả năng kết hợp các ràng buộc trong thuật toán phân cụm.

1.5 Các yêu cầu cho kỹ thuật PCDL

Hầu hết các nghiên cứu và phát triển thuật toán phân cụm dữ liệu đều nhằm thoả mãn các yêu cầu cơ bản sau:

- *Có khả năng mở rộng* (Scalability): Một số thuật toán có thể ứng dụng tốt cho tập dữ liệu nhỏ (khoảng 200 bản ghi dữ liệu) nhưng không hiệu quả khi áp dụng cho tập dữ liệu lớn (Khoảng 1 triệu bản ghi).
- *Thích nghi với các kiểu dữ liệu khác nhau:* Thuật toán có thể áp dụng hiệu quả cho việc phân cụm các tập dữ liệu với nhiều kiểu dữ liệu khác nhau như dữ liệu kiểu số, kiểu nhị phân, dữ liệu kiểu hạng mục,.. và thích nghi với kiểu dữ liệu hỗn hợp giữa các dữ liệu đơn trên.
- *Khám phá ra các cụm với hình thù bất kỳ*: do hầu hết các CSDL có chứa nhiều cụm dữ liệu với các hình thù khác nhau như: hình lõm, hình cầu, hình que, …Vì vậy, để khám phá được các cụm có tính tự nhiên thì các thuật toán phân cụm cần phải có khả năng khám phá ra các cụm có hình thù bất kỳ.
- *Tối thiểu lượng tri thức cần cho xác định các tham số vào*: do các giá trị đầu vào thường rất ảnh hưởng đến thuật toán phân cụm và rất phức tạp để xác định các giá trị vào thích hợp đối với các CSDL lớn.
- *Ít nhạy cảm với thứ tự của dữ liệu vào:* Cùng một tập dữ liệu, khi đưa vào xử lý cho thuật toán PCDL với các thứ tự vào của các đối tượng dữ liệu ở các lần thực hiện khác nhau thì không ảnh hưởng lớn đến kết quả phân cụm.
- *Khả năng thích nghi với dữ liệu nhiễu cao:* Hầu hết các dữ liệu phân cụm trong Data Mining đều chứa đựng các dữ liệu lỗi, dữ liệu không đầy đủ, dữ liệu

rác. Thuật toán phân cụm không những hiệu quả đối với các dữ liệu nhiễu mà còn tránh dẫn đến chất lượng phân cụm thấp do nhạy cảm với nhiễu.

- *Ít nhạy cảm với các tham số đầu vào:* Nghĩa là giá trị của các tham số đầu vào khác nhau ít gây ra các thay đổi lớn đối với kết quả phân cụm.
- *Thích nghi với dữ liệu đa chiều*: Thuật toán có khả năng áp dụng hiệu quả cho dữ liệu có số chiều khác nhau.
- *Dễ hiểu, cài đặt và khả dụng.*

Các yêu cầu này đồng thời là các tiêu chí để đánh giá hiệu quả của các phương pháp phân cụm dữ liệu, đây là các thách thức cho các nhà nghiên cứu trong lĩnh vực PCDL.

1.6 Giới thiệu thuật toán phân cụm dữ liệu điển hình.

Sau đây là một số họ thuật toán PCDL điển hình như: Họ các thuật toán phân cụm phân hoạch (Patitional), họ các thuật toán phân cụm phân cấp (Hierachical), họ các thuật toán phân cụm dựa trên lưới và các thuật toán PCDL đặc thù khác như: các thuật toán phân cụm dựa trên mật độ, các thuật toán phân cụm dựa trên mô hình,…

Họ các thuật toán phân hoạch

Họ các thuật toán phân cụm phân hoạch bao gồm các thuật toán được áp dụng nhiều trong thực tế như K-means, PAM (Partioning Around Medoids), CLARA (Clustering LARge Applications), CLARANS (Clustering LARge ApplicatioNS).

Thuật toán k-means

Thuật toán phân hoạch K-means do MacQeen đề xuất trong lĩnh vực thống kê năm 1967, mục đích của thuật toán k-means là sinh ra k cụm dữ liệu $\{C_1, C_2, \ldots, C_k\}$ từ một tập dữ liệu chứa n đối tượng trong không gian d chiều $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ $(i = \overline{1,n})$, sao cho hàm tiêu chuẩn: $E = \sum_{i=1}^{k} \sum_{x \in C_i} D^2(x - m_i)$ đạt giá trị tối thiểu. Trong đó: m_i là trọng tâm của cụm C_i, D là khoảng cách giữa hai đối tượng.

Trọng tâm của một cụm là một véc tơ, trong đó giá trị của mỗi phần tử của nó là trung bình cộng của các thành phần tương ứng của các đối tượng vectơ dữ liệu trong cụm đang xét. Tham số đầu vào của thuật toán là số cụm k, và tham số đầu ra của thuật toán là các trọng tâm của các cụm dữ liệu. Độ đo khoảng cách D giữa các đối tượng dữ liệu thường được sử dụng dụng là khoảng cách Euclide, bởi vì đây là mô hình khoảng cách dễ để lấy đạo hàm và xác định các cực trị tối thiểu. Hàm tiêu chuẩn và độ đo

khoảng cách có thể được xác định cụ thể hơn tuỳ vào ứng dụng hoặc các quan điểm của người dùng. Thuật toán k-means bao gồm các bước cơ bản như trong hình sau:

InPut: Số cụm k và các trọng tâm cụm $\{m_j\}^k_{j=1}$;

OutPut: Các cụm C_i ($i = \overline{1,k}$) và hàm tiêu chuẩn E đạt giá trị tối thiểu;

Begin

Bước 1: Khởi tạo:

Chọn k trọng tâm $\{m_j\}^k_{j=1}$ ban đầu trong không gian R^d (d là số chiều của dữ liệu). Việc lựa chọn này có thể là ngẫu nhiên hoặc theo kinh nghiệm.

Bước 2: Tính toán khoảng cách:

Đối với mỗi điểm X_i (1<=i<=n), tính toán khoảng cách của nó tới mỗi trọng tâm m_j j=1,k. Và sau đó tìm trọng tâm gần nhất đối với mỗi điểm.

Bước 3: Cập nhật lại trọng tâm:

Đối với mỗi j=1,k, cập nhật trọng tâm cụm m_j bằng các xác định trung bình cộng của các vectơ đối tượng dữ liệu.

Bước 4: Điều kiện dừng

Lặp các bước 2 và 3 cho đến khi các trọng tâm của cụm không thay đổi.

End.

Hình: Các bước thực hiện của thuật toán k-means

K-means biểu diễn các cụm bởi các trọng tâm của các đối tượng trong cụm đó. do k-means phân tích phân cụm đơn giản nên có thể áp dụng đối với tập dữ liệu lớn. Tuy nhiên, nhược điểm của k-means là chỉ áp dụng với dữ liệu có thuộc tính số và khám ra các cụm có dạng hình cầu, k-means còn rất nhạy cảm với nhiễu và các phần tử ngoại lai trong dữ liệu.

Hơn nữa, chất lượng phân cụm dữ liệu của thuật toán k-means phụ thuộc nhiều vào các tham số đầu vào như: số cụm k và k trọng tâm khởi tạo ban đầu. Trong trường hợp, các trọng tâm khởi tạo ban đầu mà quá lệch so với các trọng tâm cụm tự nhiên thì kết quả phân cụm của k-means là rất thấp, nghĩa là các cụm dữ liệu được khám phá rất lệch so với các cụm trong thực tế. Trên thực tế người ta chưa có một giải pháp tối ưu nào để chọn các tham số đầu vào, giải pháp thường được sử dụng nhất là thử nghiệm với các giá trị đầu vào k khác nhau rồi sau đó chọn giải pháp tốt nhất.

Đến nay, đã có rất nhiều thuật toán kế thừa tư tưởng của thuật toán k-means áp dụng trong Data Mining để giải quyết với tập dữ liệu có kích thước rất lớn đang được áp dụng rất hiệu quả và phổ biến như thuật toán k-modes, PAM, CLARA, CLARANS, k- prototypes, …

Hạn chế chung của các thuật toán phân cụm phân hoạch là chỉ thích hợp đối với dữ liệu số và ít chiều, và chỉ khám phá ra các cụm dạng hình cầu, thế nhưng chúng lại áp dụng tốt với dữ liệu có các cụm phân bố độc lập và trong mỗi cụm có mật độ phân bố cao.

1.7 Bài toán phân cụm dữ liệu

Bài toán phân cụm dữ liệu thường được hiểu là một bài toán học không giám sát và được phát biểu như sau.

Cho tập N đối tượng dữ liệu $X=\{x_1,...,x_N\}$ (bài này ta hạn chế chỉ xét các đổi tượng trong không gian số học n-chiều: $x_i \in R^n$), ta cần chia X thành các cụm đôi một không giao nhau: $X=\bigcup_{i=1}^{k} C_i$ sao cho các đối tượng trong cùng một cụm C_i thì tương tự nhau và các đối tượng trong các cụm khác nhau thì khác nhau hơn theo một cách nhìn nào đó. Số lượng k các cụm có thể cho trước hoặc xác định nhờ phương pháp phân cụm. Để thực hiện phân cụm, ta cần xác định được mức độ tương tự giữa các đối tượng, tiêu chuẩn để phân cụm, trên cơ sở đó xây dựng mô hình và các thuật toán phân cụm theo nhiều cách tiếp cận. Mỗi cách tiếp cận cho ta kết quả phân cụm với ý nghĩa sử dụng khác nhau.

Chương 2 HỆ QUẢN TRỊ CSDL ORACLE

2.1 Giới thiệu Oracle

Oracle bao gồm một tập hợp hoàn thiện các sản phẩm xây dựng ứng dụng và người dùng cuối được trang bị các giải pháp kỹ thuật thông tin hoàn hảo. Các ứng dụng Oracle tương thích với hầu hết các hệ điều hành từ các máy tính cá nhân đến các hệ thống xử lý song song lớn.

Oracle cung cấp một hệ quản trị cơ sở dữ liệu (Database Management System - DBMS) uyển chuyển: Oracle Server để lưu giữ và quản lý các thông tin dùng trong các ứng dụng. Oracle Server là một hệ quản trị CSDL điều khiển:

Các sản phẩm của Oracle bao gồm:

- Oracle TextRetrieval
- Pro* ORACLE
- Oracle Card
- Oracle CASE
- SQL * Plus
- SQL, SQL * Plus và PL/SQL là các đặc tính của Oracle.

➢ **SQL:**

- Là ngôn ngữ dùng để truy xuất cơ sở dữ liệu quan hệ, kể cả Oracle.
- Có thể được dùng với mỗi công cụ Oracle khi có yêu cầu truy xuất dữ liệu.

➢ **PL/SQL:**

- Là ngôn ngữ thủ tục Oracle để viết các ứng dụng luận lý và thao tác dữ liệu bên ngoài CSDL.
- Có thể bao gồm một tập con các lệnh SQL khi có yêu cầu truy xuất dữ liệu.
- Sẵn có trong Oracle Server.

➢ **SQL * Plus:**

- Là sản phẩm Oracle trong đó có thể dùng cả SQL và PL/SQL.
- Còn có các ngôn ngữ lệnh riêng để điều khiển hành vi của sản phẩm và định dạng output từ các truy vấn SQL.

2.2 Cấu trúc cơ sở dữ liệu (CSDL):

- **Tablespaces & Data files:**

Một CSDL Oracle được lưu giữ trong một hoặc nhiều đơn vị nhớ logic gọi là tablespace.

Mỗi tablespace được lưu giữ thành một hoặc nhiều file vật lý gọi là Data file.

- **Schema Objects (đối tượng CSDL):**

Schema Objects là cấu trúc logic lưu trữ dữ liệu. Schema Objects bao gồm tables, views, sequences, synonyms, indexes, clusters, database links, procedures, packages.

- **Tables:**

Là đơn vị nhỏ nhất của việc lưu trữ dữ liệu trong một CSDL Oracle. Dữ liệu được lưu thành dòng và cột. Mỗi table được định nghĩa bằng 1 table name và tập các cột. Mỗi cột (field) có một tên cột, kiểu, và độ lớn. Mỗi dòng là tập hợp những thông tin của các cột gọi là 1 record.

- **Views:**

View tương tự như Window mà qua đó dữ liệu trong table có thể được xem hoặc biến đổi. Một view chỉ được lưu giữ dưới dạng câu lệnh SELECT. View là một dạng table ảo nghĩa là table không thực sự tồn tại mà chỉ xuất hiện để user xem. Một view tự nó không có dữ liệu mà sẽ tương tác dữ liệu dựa vào các table cơ sở.

View giới hạn việc xâm nhập dữ liệu, cho phép các users nhập các câu truy vấn đơn giản để lấy kết quả từ các câu truy vấn phức tạp thông qua 1 view, cung cấp dữ liệu độc lập cho nhiều người sử dụng hay các chương trình ứng dụng.

View được chia thành 2 loại: view đơn và view phức. View đơn là view chỉ truy xuất dữ liệu từ 1 table và không chứa bất kỳ hàm hay GROUP dữ liệu nào, ngược lại gọi là view phức.

- **Synonyms:**

Synonym là một bí danh của một table, view, sequence, procedure, function hay một package. Synonym được sử dụng cho việc bảo mật và tiện lợi trong truy xuất dữ liệu. Có 2 loại synonym: Public và Private.

- **Index (chỉ mục):**

Index của Bảng được tạo ra nhằm tăng tốc độ truy xuất, tăng hiệu quả của tính duy nhất trên một hoặc một tập của cột.

2.3 Sử dụng phân cụm (CLUSTERING) trong Oracle

Phân cụm trong Oracle được thực hiện với thủ tục CTX_CLS.CLUSTERING.

CTX_CLS.CLUSTERING chỉ định đầu ra cho 2 bảng:

- một bảng chỉ định hiển thị 1 tài liệu,tuy nhiên các cụm tài liệu thường thì giống nhau

, thông tin được lấy từ văn bản, từ các cụm, và từ nhiều điểm tương tự giữa tài liệu và cụm.

- một bảng mô tả thông tin về cụm, về những cụm giống nhau, chứa đựng những cụm thông tin nhận dạng, các dòng văn bản được mô tả bởi các cụm,gán nhãn cho các cụm và nâng cao khả năng cho các cụm.

CTX_CLS.CLUSTERING còn sử dụng thuật toán KMEAN_CLUSTERING để thực hiện việc phân cụm. Sử dụng KMEAN_CLUSTERING để xác định công việc cho CTX_CLS.CLUSTERING.

Gói phần mềm này CTX_CLS.CLUSTERING cho phép bạn thực hiện phân loại tài liệu

KMEAN_CLUSTERING có những thuộc tính sau

Tên thuộc tính	Kiểu	Data Mặc định	Min giá trị gia tăng	Max	Mô tả
MAX_DOCTERMS	I	50	10	8192	Chỉ định tối đa số điều khoản khác biệt đại diện cho 1trong những tài liệu.
MAX_FEATURES	I	3,000	1	500,000	Chỉ định tối đa số lượng các tính năng khác biệt
THEME_ON	B	FALSE	NULL	NULL	True chỉ định để sử dụng các chủ đề như là các điểm đặc biệt. Clustering(phân cụm) với chủ đề yêu cầu có 1 cài đặt cơ bản cơ sở. .
TOKEN_ON	B	TRUE	NULL	NULL	Chỉ định TRUE để sử dụng các dấu hiệu đặc biệt , đặc trưng trong nội dung
STEM_ON	B	FALSE	NULL	NULL	Chỉ định True để sử dụng các dấu hiệu trong nội dung phần thân . Chỉ làm việc khi chuyển hướng Index_stem các lexer.
MEMORY_SIZE	I	500	10	4000	Chỉ định chính xác kích cỡ bộ nhớ,

trong MB.

SECTION_WEIGHT 1 2 0 100 Ghi rõ sự xuất hiện hệ số để khi thêm 1 thuật ngữ về 1chủ đề, lĩnh vực nào đó thì cũng được coi là 1 phần bình thường của giới hạn. VD ngầm định thuật ngữ Cat (con mèo)"<A>cat</A>. Cat là một lĩnh vực và được xem như là bình thường với 1 thuật ngữ xuất hiện 2 lần nhưng bạn có thể chỉ ra rằng nó được xử lý như là 1 thuật ngữ bình thường với 1 trọng lượng lên đến 100. Nhóm_trọng lượng chỉ có ý nghĩa khi các chỉ mục chính xác định 1 phần lĩnh vực.

CLUSTER_NUM I 200 2 20000 Xác định tổng số cụm được tạo ra.

- Sử dụng thủ tục CTX_CLS.CLUSTERING này *để phân cụm một tập hợp những tài liệu, 1 cụm, 1 nhóm những tài liệu tương tự nhau trong nội dung*. ví dụ sau: nếu các đơn đặt hàng chứa hầu hết số mặt hàng như nhau khi đó cluster có thể có ích. Nếu các đơn đặt hàng có chỉ một hay hai mặt hàng trong khi đó các đơn đặt hàng khác có hàng trăm mặt hàng trong trường hợp này sử dụng cluster là không thích hợp.

Một tập hợp kết quả phân cụm bao gồm:

Những tài liệu được chỉ định và các cụm đã được mô tả, tài liệu được chỉ định với kết quả thiết lập hình thức cho các tài liệu liên quan, tập hợp kết quả mô tả cụm chứa thông tin về 1 cụm chủ đề nào đó. Đây là kết quả của phân cụm, các cụm dòng text được mô tả, và gán nhãn cho các cụm, gán điểm số cao cho các cụm tài liệu. Các cụm đuợc xuất ra có thứ tự. Những tài liệu có nhiều điểm giống nhau thì đươc cho điểm (Xem VD dưới). Việc sản sinh nhiều cụm hơn yêu cầu nhiều thời gian tính toán hơn.

Bạn giới hạn cho những cụm phát sinh thêm bằng thuộc tính CLUSTER_NUM của thuât toán KMEAN_CLUSTERING.

chú ý: những thuộc tính sử dụng để xác định những cụm có thể gồm những từ đơn giản

Những kiểu sử dụng để tạo ra sự ưu tiên cho thủ tục CTX_CLS.CLUSTERING

Cú pháp: **Table Result Set (**Bảng kết quả **)**

ctx_cls.clustering (index_name IN VARCHAR2, docid IN VARCHAR2,

doctab_name IN VARCHAR2, clstab_name IN VARCHAR2, pref_name IN VARCHAR2 DEFAULT NULL);

index_name

Tên đặc biệt của cái giá trị chọn lọc trong bảng

docid

Chỉ rõ tên cột ID tài liệu của bảng chọn

doctab_name

Tên đặc biệt của văn bản được gắn với tên bảng. Đây là thủ tục để tạo bảng với cấu trúc tiếp theo:

doc_assign(docid number, clusterid number, score number);

Mô tả cột

DOCID: ID của tài liệu để nhận ra tài liệu.

CLUSTERID: ID cuả 1 cụm liên quan đến tài liệu. Nếu CLUSTERID là -1, thì cụm chứa tài liệu "hỗn hợp "; VD, không thể chỉ định cụm khác trong danh mục tài liệu, nhiều điểm sẽ được kết hợp giữa cụm và văn bản

clstab_name

Chỉ rõ tên của cụm được mô tả trong bảng. Đây là thủ tục để tạo bảng với cấu trúc tiếp theo:

cluster_desc(

clusterid NUMBER,

descript VARCHAR2(4000),

label VARCHAR2(200),

sze NUMBER,

quality_score NUMBER,

parent NUMBER);

Cột Mô tả

CLUSTERID Cluster ID để nhận biết các cụm. Nếu CLUSTERID là -1, sau đó nhóm có chứa "miscellaneous" tài liệu, ví dụ, các tài liệu có thể không được xếp vào nhóm nào khác thể loại.

Chuôi DESCRIPT để mô tả các cụm.

Một nhãn LABEL đề nghị cho các nhóm.

SiZE tham số này hiện nay không có giá trị.

QUALITY_SCORE Các điểm chất lượng của các nhóm. Một số cao hơn cho biết tính mạch lạc hơn.

PHỤ HUYNH Các nhóm phụ huynh id. Không có nghĩa là không có nhóm phụ huynh. Nếu bạn cần thêm các cột, bạn có thể tạo bảng, trước khi bạn gọi thủ tục này.

pref_name

chỉ rõ những tên ưu tiên

Cú pháp: (In-Memory Result Set) Tập hợp các kết quả đã lưu

Bạn có thể đặt bộ kết quả trong cấu trúc bộ nhớ làm tăng hiệu suất. 2 trong số những bảng lưu được định nghĩa trong gói CTX_CLS package cho văn bản được chỉ định và những cụm được mô tả.

```
CTX_CLS.CLUSTERING(
index_name     IN VARCHAR2,
 docid          IN VARCHAR2,
 dids           IN DOCID_TAB,
 doctab_name    IN OUT NOCOPY DOC_TAB,
 clstab_name    IN OUT NOCOPY CLUSTER_TAB,
 pref_name      IN VARCHAR2  DEFAULT NULL
 );
```

index_name (danh mục tên)

Chỉ rõ danh mục tên của tình huống chọn lọc trong bảng

docid

Chỉ rõ tên cột ID tài liệu của bảng chọn

dids

Chỉ rõ tên của bộ nhớ trong docid_tab.

TYPE docid_tab IS TABLE OF number INDEX BY BINARY_INTEGER;

Kiểu docid_tab là bảng của chỉ số nhị phân _ nguyên

doctab_name

chỉ rõ tên của tài liệu được ấn định trong bộ nhớ. tiếp theo là định nghĩa bảng:

```
TYPE doc_rec IS RECORD (
docid NUMBER,
clusterid NUMBER,
score NUMBER )
```

TYPE doc_tab IS TABLE OF doc_rec INDEX BY BINARY_INTEGER;

Mô tả cột

DOCID để xác định tài liệu

CLUSTERID Nhận dạng cụm. Nếu CLUSTERID là -1, thì cụm chứa tài liệu "hỗn hợp "; Nó là: tài liệu không thể được ấn định trong cụm khác

cls_tab

Ghi rõ tên của cụm được mô tả trong bộ nhớ của bảng

TYPE cluster_rec IS RECORD(

clusterid NUMBER,

descript VARCHAR2(4000),

label VARCHAR2(200),

sze NUMBER,

quality_score NUMBER,

parent NUMBER);

TYPE cluster_tab IS TABLE OF cluster_rec INDEX BY BINARY_INTEGER;

Mô tả

CLUSTERID Nhận dạng cụm. Nếu CLUSTERID

là -1, thì cụm chứa tài liệu "hỗn hợp "; VD, tài liệu không thể được ấn định trong danh mục cụm khác.

DESCRIPT chuỗi mô tả cụm.

LABEL gán nhãn cho cụm.

SZE Tham số này hiện thời không có giá trị.

QUALITY_SCORE những cụm đạt chất lượng. 1 số điểm lớn được biểu thị khi nó gắn với nhau.

PARENT Cụm ID gốc. Zero không có ý nghĩa với cụm gốc

pref_name tên ưa dùng. cho thuộc tính cụm của tài liệu

2. Oracle đã tích hợp sẵn các thuật toán Phân cụm như K_mean... vào bộ Oracle 10i nên ta chỉ tận dụng nó thôi. Với 1 dữ liệu trong Oracle có số dòng và bảng rất lớn thì việc tính toán rất mất thời gian và chi phí nên cần thiết phải phân cụm.

2.4 Phân loại tài liệu văn bản trong Oracle

Phân loại không giám sát (Unsupervised Clustering)

Một vấn đề lớn đối mặt các doanh nghiệp và tổ chức trong ngày hôm nay là thông tin quá tải. Phân loại ra khỏi các tài liệu hữu ích từ các tài liệu không được quan tâm là vấn đề được đặt ra cho cá nhân và tổ chức.

Một cách để phân loại là : thông qua nhiều tài liệu và sử dụng công cụ tìm kiếm từ khóa. Tuy nhiên, từ khóa tìm kiếm có các hạn chế. Một trong những mặt hạn chế chính là các từ khóa tìm kiếm không phân biệt được các ngữ cảnh khác nhau. Trong nhiều ngôn ngữ, một từ hoặc cụm từ có thể có nhiều ý nghĩa, do đó, một kết quả tìm kiếm có thể ở nhiều kết quả phù hợp không được mong muốn trên chủ đề. Ví dụ, một yêu cầu tìm kiếm về ngân hàng (river bank), cụm từ ngân hàng có thể trả lại các tài liệu về các sông Hudson & Đúng phải là Ngân hàng Công ty, bởi vì từ ngân hàng có hai ý nghĩa.

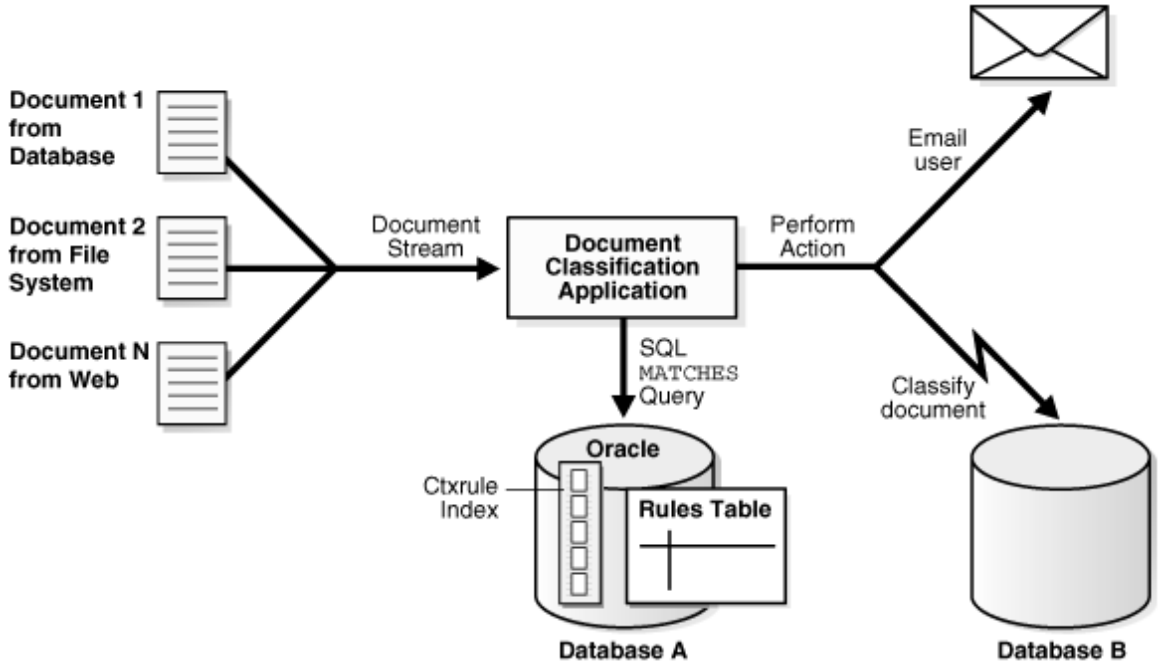
Một chiến lược thay thế là có con người thông qua phân loại các tài liệu và phân loại nội dung của chúng, nhưng điều này là không khả thi đối với số lượng rất lớn các tài liệu.

Oracle Text cung cấp phương pháp tiếp cận khác nhau để phân loại tài liệu. Theo quy định trên cơ sở phân loại, bạn viết các quy định phân loại cho mình. Với giám sát phân loại, Oracle tạo ra các văn bản quy định phân loại dựa trên một bộ các mẫu văn bản mà bạn trước khi phân loại. Cuối cùng, với phân cụm không có giám sát, Oracle tất cả các văn bản thực hiện các bước, từ văn bản quy định việc phân loại để phân loại các tài liệu, cho bạn.

Phân loại ứng dụng

Oracle Văn bản cho phép bạn để xây dựng tài liệu phân loại ứng dụng. Một tài liệu phân loại ứng dụng thực hiện một số hành động dựa trên các tài liệu nội dung. Bao gồm các hành động phân loại id vào một tài liệu để tra cứu trong tương lai hoặc gửi tài liệu đến một người dùng. Kết quả là một thiết lập hoặc dòng của phân loại tài liệu.
Hình 6-1 minh họa cách thức phân loại quá trình làm việc.

Oracle Text cho phép bạn tạo các tài liệu phân loại ứng dụng trong nhiều cách khác nhau. Chương này xác định một điển hình phân loại kịch bản và hiển thị như thế nào bạn có thể sử dụng Oracle Text để xây dựng một giải pháp.



Hình 6-1 Tổng quan về một tài liệu phân loại ứng dụng

Oracle Văn bản cho phép bạn phân loại các tài liệu trong các cách sau:

Phân loại không giám sát (supervised clustering). Tất cả các bước từ nhóm các tài liệu của bạn vào danh mục các văn bản quy định là tự động với CTX_CLS.CLUSTERING. Oracle Text phân tích thống kê tài liệu của bạn thiết lập và kết hợp chúng với cụm theo nội dung.

Ưu điểm:

- Bạn không cần phải cung cấp các quy tắc phân loại hoặc các tài liệu như là một mẫu đào tạo thiết lập.
- Giúp để khám phá các mẫu và nội dung tương tự trong tài liệu của bạn thiết lập mà bạn có thể mở ra.
- Trong thực tế, bạn có thể sử dụng phân loại không giám sát khi bạn không có một ý tưởng rõ ràng về những quy tắc phân loại. Một trong những kịch bản có thể được sử dụng để phân loại không giám sát là đầu tiên cung cấp một tập hợp các chuyên mục, quy tắc, và sau đó xây dựng trên các giám sát thông qua các phân loại.

Nhược điểm:

- Clustering có thể cho kết quả bất ngờ nhất, clustering hoạt động không phải là người dùng xác định, nhưng dựa trên thuật toán nội bộ.
- Bạn không nhìn thấy rằng các quy tắc tạo ra cụm.

- Clustering cần nhiều các hoạt động của CPU nên có thể mất ít nhất là trong cùng thời gian như lập chỉ mục.

Chương 3 MÔ HÌNH USE CASE

3.1 Giới thiệu Use Case trong phân tích thiết kế hướng đối tượng

Trong giai đoạn phân tích, người sử dụng cộng tác cùng nhóm phát triển phần mềm tạo nên một tổ hợp thông tin quan trọng về yêu cầu đối với hệ thống. Không chỉ là người cung cấp thông tin, bản thân người sử dụng còn là một thành phần hết sức quan trọng trong bức tranh toàn cảnh đó và nhóm phát triển cần phải chỉ ra được phương thức hoạt động của hệ thống tương lai theo hướng nhìn của người sử dụng. Như vậy công cụ giúp ta mô hình hoá hệ thống từ hướng nhìn của người sử dụng gọi là Use Case.

Use Case là một công cụ trợ giúp cho công việc của nhà phân tích cùng người sử dụng quyết định tính năng của hệ thống. Một tập hợp các Use Case sẽ làm nổi bật một hệ thống theo phương diện những người dùng định làm gì với hệ thống này.

Nhìn chung, có thể coi một Use case như là tập hợp của một loạt các cảnh kịch về việc sử dụng hệ thống. Mỗi cảnh kịch mô tả một chuỗi các sự kiện. Mỗi một chuỗi này sẽ được kích hoạt bởi một người nào đó, một hệ thống khác hay là một phần trang thiết bị nào đó, hoặc là một chuỗi thời gian. Những thực thể kích hoạt nên các chuỗi sự kiện như thế được gọi là các **Tác Nhân (Actor)**. Kết quả của chuỗi này phải có giá trị sử dụng đối với hoặc là tác nhân đã gây nên nó hoặc là một tác nhân khác.

3.2 Mô hình hóa Use Case

Trường hợp sử dụng là một kỹ thuật mô hình hóa được sử dụng để mô tả một hệ thống mới sẽ phải làm gì hoặc một hệ thống đang tồn tại làm gì. Một mô hình Use Case được xây dựng qua một quá trình mang tính vòng lặp (interative), trong đó những cuộc hội thảo bàn luận giữa nhóm phát triển hệ thống và khách hàng (hoặc/và người sử dụng cuối) sẽ dẫn tới một đặc tả yêu cầu được tất cả mọi người chấp nhận. Người cha tinh thần của mô hình hóa Use Case là Ivar Jacobson, ông đã tạo nên kỹ thuật mô hình hóa dựa trên những kinh nghiệm thu thập được trong quá trình tạo hệ thống AXE của hãng Erisson. Use Case đã nhận được một sự quan tâm đặc biệt lớn lao từ phía cộng đồng hướng đối tượng và đã tác động lên rất nhiều phương pháp hướng đối tượng khác nhau.

Những thành phần quan trọng nhất của một mô hình Use Case là Use Case, tác nhân và hệ thống. Ranh giới của hệ thống được định nghĩa qua chức năng tổng thể mà

hệ thống sẽ thực thi. Chức năng tổng thể được thể hiện qua một loạt các Use Case và mỗi một Use Case đặc tả một chức năng trọn vẹn, có nghĩa là Use Case phải thực thi toàn bộ chức năng đó, từ sự kiện được kích hoạt đầu tiên bởi một tác nhân ngoại cảnh cho tới khi chức năng đòi hỏi được thực hiện hoàn tất. Một Use Case luôn luôn phải cung cấp một giá trị nào đó cho một tác nhân, giá trị này là những gì mà tác nhân mong muốn từ phía hệ thống. Tác nhân là bất kỳ một thực thể ngoại cảnh nào mong muốn tương tác với hệ thống. Thường thường, đó là một người sử dụng của hệ thống, nhưng nhiều khi cũng có thể là một hệ thống khác hoặc là một dạng máy móc thiết bị phần cứng nào đó cần tương tác với hệ thống.

Mục tiêu chính yếu đối với các Use Case là:

- Để quyết định và mô tả các yêu cầu về mặt chức năng của hệ thống, đây là kết quả rút ra từ sự thỏa thuận giữa khách hàng (và/hoặc người sử dụng cuối) và nhóm phát triển phần mềm.

- Để tạo nên một lời mô tả rõ ràng và nhất quán về việc hệ thống cần phải làm gì, làm sao để mô hình có thể được sử dụng nhất quán suốt toàn bộ quá trình phát triển, được sử dụng làm công cụ giao tiếp cho tất cả những người phát triển nên các yêu cầu này, và để tạo nên một nền tảng cho việc tạo nên các mô hình thiết kế cung cấp các chức năng được yêu cầu.

- Để tạo nên một nền tảng cho các bước thử nghiệm hệ thống, đảm bảo hệ thống thỏa mãn đúng những yêu cầu do người sử dụng đưa ra. Trong thực tế thường là để trả lời câu hỏi: Liệu hệ thống cuối cùng có thực hiện những chức năng mà khởi đầu khách hàng đã đề nghị?

- Để cung cấp khả năng theo dõi các yêu cầu về mặt chức năng được chuyển thành các lớp cụ thể cũng như các thủ tục cụ thể trong hệ thống.

- Để đơn giản hóa việc thay đổi và mở rộng hệ thống qua việc thay đổi và mở rộng mô hình Use Case, sau đó chỉ theo dõi riêng những Use Case đã bị thay đổi cùng những hiệu ứng của chúng trong thiết kế hệ thống và xây dựng hệ thống.

Những công việc cụ thể cần thiết để tạo nên một mô hình Use Case bao gồm:

1. Định nghĩa hệ thống (xác định phạm vi hệ thống)
2. Tìm ra các tác nhân cũng như các Use Case
3. Mô tả Use Case
4. Định nghĩa mối quan hệ giữa các Use Case

5. Kiểm tra và phê chuẩn mô hình.

Đây là một công việc mang tính tương tác rất cao, bao gồm những cuộc thảo luận với khách hàng và những người đại diện cho các loại tác nhân. Mô hình Use Case bao gồm các biểu đồ Use Case chỉ ra các tác nhân, Use Case và mối quan hệ của chúng với nhau. Các biểu đồ này cho ta một cái nhìn tổng thể về mô hình, nhưng những lời mô tả thực sự của từng Use Case thường lại là văn bản. Vì các mô hình trực quan không thể cung cấp tất cả các thông tin cần thiết, nên cần thiết phải dùng cả hai kỹ thuật trình bày đó.

Có rất nhiều người quan tâm đến việc sử dụng các mô hình Use Case. Khách hàng (và/hoặc người sử dụng cuối) quan tâm đến chúng vì mô hình Use Case đặc tả chức năng của hệ thống và mô tả xem hệ thống có thể và sẽ được sử dụng ra sao. Các Use Case vì vậy phải được mô tả trong những thuật ngữ và ngôn ngữ của khách hàng/người sử dụng.

Nhà phát triển cần đến các mô hình Use Case để hiểu hệ thống cần phải làm gì, và qua đó có được một nền tảng cho những công việc tương lai (các mô hình khác, các cấu trúc thiết kế và việc thực thi xây dựng hệ thống bằng code).

Các nhóm chuyên gia thử nghiệm tích hợp và thử nghiệm hệ thống cần đến Use Case để thử nghiệm và kiểm tra xem hệ thống có đảm bảo sẽ thực hiện đúng chức năng đã được đặc tả trong giai đoạn đầu.

Và cuối cùng, bất kỳ người nào liên quan đến những hoạt động liên kết đến chức năng của hệ thống đều có thể quan tâm đến các mô hình Use Case; ví dụ như các nhóm tiếp thị, bán hàng, hỗ trợ khách hàng và các nhóm soạn thảo tài liệu.

Mô hình Use Case mô tả hướng nhìn Use Case của hệ thống. Hướng nhìn này là rất quan trọng, bởi nó ảnh hưởng đến tất cả các hướng nhìn khác của hệ thống. Cả cấu trúc logic lẫn cấu trúc physic đều chịu ảnh hưởng từ các Use Case, bởi chức năng được đặc tả trong mô hình này chính là những chức năng được thực thi trong các cấu trúc kia. Mục đích cuối cùng là thiết kế ra một giải pháp thỏa mãn các yêu cầu đó.

Mô hình hóa các Use Case chẳng phải chỉ được dùng để nắm bắt các yêu cầu của hệ thống mới; nó cũng còn được sử dụng để hỗ trợ cho việc phát triển một phiên bản mới của hệ thống. Khi phát triển một phiên bản mới của hệ thống đang tồn tại, người ta sẽ bổ sung thêm các chức năng mới vào mô hình Use Case đã có bằng cách

thêm vào các tác nhân mới cũng như các Use Case mới, hoặc là thay đổi đặc tả của các Use Case đã có. Khi bổ sung thêm vào mô

hình Use Case đang tồn tại, hãy chú ý để không bỏ ra bất kỳ một chức năng nào vẫn còn được cần tới.

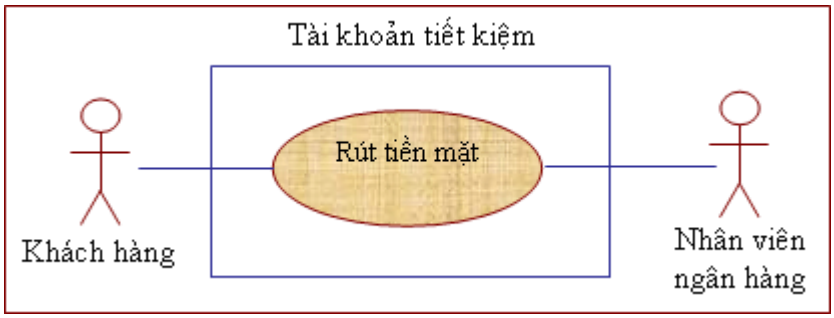
3.3 Biểu đồ Use Case

Biểu đồ Use Case (Use Case Diagram).

Tóm tắt: Một biểu đồ Use Case thể hiện:

- Hệ thống
- Tác nhân
- Use Case.

Ví dụ biểu đồ Use Case trong UML:



Hình 4.1- Một ví dụ biểu đồ Use case trong UML

Trong đó:

- Hệ thống được thể hiện qua hình chữ nhật với tên hệ thống ở bên trên
- Tác nhân được thể hiện qua kí hiệu hình nhân
- Use Case được thể hiện qua hình ellipse

3.4 Quan hệ giữa các Use Case

Có ba loại quan hệ Use Case: Quan hệ mở rộng, quan hệ sử dụng và quan hệ tạo nhóm.

3.4.1 Miêu tả Use Case

Như đã trình bày, lời miêu tả một Use Case thường được thực hiện trong văn bản. Đây là lời đặc tả đơn giản và nhất quán về việc các tác nhân và các Use Case (hệ thống) tương tác với nhau ra sao. Nó tập trung vào ứng xử đối ngoại của hệ thống và không đề cập tới việc thực hiện nội bộ bên trong hệ thống. Ngôn ngữ và các thuật ngữ

được sử dụng trong lời miêu tả chính là ngôn ngữ và các thuật ngữ được sử dụng bởi khách hàng/người dùng.

Văn bản miêu tả cần phải bao gồm những điểm sau:

- Mục đích của Use Case: Mục đích chung cuộc của Use Case là gì? Cái gì cần phải được đạt tới? Use Case nói chung đều mang tính hướng mục đích và mục đích của mỗi Use Case cần phải rõ ràng.

- Use Case được khởi chạy như thế nào: Tác nhân nào gây ra sự thực hiện Use Case này? Trong hoàn cảnh nào?

- Chuỗi các thông điệp giữa tác nhân và Use Case: Use Case và các tác nhân trao đổi thông điệp hay sự kiện nào để thông báo lẫn cho nhau, cập nhật hoặc nhận thông tin và giúp đỡ nhau quyết định? Yếu tố nào sẽ miêu tả dòng chảy chính của các thông điệp giữa hệ thống và tác nhân, và những thực thể nào trong hệ thống được sử dụng hoặc là bị thay đổi?

- Dòng chảy thay thế trong một Use Case: Một Use Case có thể có những dòng thực thi thay thế tùy thuộc vào điều kiện. Hãy nhắc đến các yếu tố này, nhưng chú ý đừng miêu tả chúng quá chi tiết đến mức độ chúng có thể "che khuất" dòng chảy chính của các hoạt động trong trường hợp căn bản. Những động tác xử lý lỗi đặc biệt sẽ được miêu tả thành các Use Case khác.

- Use Case sẽ kết thúc với một giá trị đối với tác nhân như thế nào: Hãy miêu tả khi nào Use Case được coi là đã kết thúc, và loại giá trị mà nó cung cấp đến tác nhân.

Hãy nhớ rằng lời miêu tả này sẽ xác định những gì được thực thi có liên quan đến tác nhân bên ngoài, chứ không phải những sự việc được thực hiện bên trong hệ thống. Văn bản phải rõ ràng, nhất quán, khiến cho khách hàng có thể dễ dàng hiểu và thẩm tra chúng (để rồi đồng ý rằng nó đại diện cho những gì mà anh/cô ta muốn từ phía hệ thống). Tránh dùng những câu văn phức tạp, khó diễn giải và dễ hiểu lầm.

Một Use Case cũng có thể được miêu tả qua một biểu đồ hoạt động. Biểu đồ hoạt động này chỉ ra chuỗi các hành động, thứ tự của chúng, các quyết định chọn lựa để xác định xem hành động nào sau đó sẽ được thực hiện.

Sau khi các Use Case đã được miêu tả, một hoạt động và một công việc đặc biệt cần phải thực hiện là thẩm tra xem các mối quan hệ có được nhận diện không. Trước khi tất cả các Use Case được miêu tả, nhà phát triển chưa thể có được những kiến thức hoàn tất và tổng thể để xác định các mối quan hệ thích hợp, thử nghiệm làm theo

phương thức đó có thể sẽ dẫn đến một tình huống nguy hiểm. Trong thời gian thực hiện công việc này, hãy trả lời các câu hỏi sau:

- Tất cả các tác nhân liên quan đến một Use Case có mối liên kết giao tiếp với Use Case đó không?

- Có tồn tại những sự tương tự giữa một loạt các tác nhân minh họa một vai trò chung và nhóm này liệu có thể được miêu tả là một lớp tác nhân căn bản (base class)?

- Có tồn tại những sự tương tự giữa một loạt các Use Case, minh họa một dòng chảy hành động chung? Nếu có, liệu điều này có thể được miêu tả là một mối quan hệ sử dụng đến với một Use Case khác?

- Có tồn tại những trường hợp đặc biệt của một Use Case có thể được miêu tả là một mối quan hệ mở rộng?

- Có tồn tại một tác nhân nào hay một Use Case nào không có mối liên kết giao tiếp? Nếu có, chắc chắn ở đây đã có chuyện lầm lạc, sai trái: Tại sao lại xuất hiện tác nhân này?

- Có lời yêu cầu nào về chức năng đã được xác định, nhưng lại không được bất kỳ một Use Case nào xử lý? Nếu thế, hãy tạo một Use Case cho yêu cầu đó.

Văn bản miêu tả một Use Case đơn giản:

Ví dụ Use Case "Cung Cấp Thông Tin Về Một Tài Khoản Tại Nhà Băng ABC": Sau khi phân tích hệ thống, ta nhận thấy cần có một Use Case để in lên màn hình của nhân viên nhà băng tất cả những chi tiết về một tài khoản của một khách hàng.

Đặc tả Use Case:

Chi tiết tài khoản: *// tên Use Case*

Số Use Case: UCSEC35

Miêu tả ngắn: *// miêu tả ngắn gọn Use Case*

Dòng chảy các sự kiện: *// dòng logic chung*

Dòng hành động chính: *// dòng logic chi tiết.*

Dòng hành động thay thế: *// chuỗi logic thay thế*

Điều kiện thoát: *// Use Case kết thúc như thế nào?*

Các yêu cầu đặc biệt: *// các yêu cầu đặc biệt*

Điều kiện trước đó: *// điều xảy ra trước khi Use Case được thực hiện*

Điều kiện sau đó: *// điều gì xảy ra sau khi Use Case được thực hiện?*

3.4.2 Thử nghiệm Use Case

Một trong các mục đích chính của Use Case là thử nghiệm (testing). Có hai loại thử nghiệm khác nhau được thực hiện ở đây: **kiểm tra** (verification) và **phê duyệt xác nhận** (validation). Kiểm tra đảm bảo là hệ thống đã được phát triển đúng đắn và phù hợp với các đặc tả đã được tạo ra. Phê duyệt xác nhận đảm bảo rằng hệ thống sẽ được phát triển chính là thứ mà khách hàng hoặc người sử dụng cuối thật sự cần đến.

Công việc **phê duyệt xác nhận** được thực hiện kề trước giai đoạn phát triển. Ngay khi một mô hình Use Case được hoàn tất (hay thậm chí có thể đang trong giai đoạn phát triển), mô hình này phải được trình bày và thảo luận với khách hàng cũng như người sử dụng. Họ cần phải xác nhận rằng mô hình này là đúng đắn, hoàn tất và thỏa mãn sự mong đợi của họ đối với hệ thống; đặc biệt là phương cách mà hệ thống cung cấp chức năng cho họ. Để làm điều đó, nhà phát triển phải đảm bảo rằng khách hàng thật sự hiểu được mô hình và ý nghĩa của chúng, để tránh trường hợp tạo ra những thứ không thể chấp nhận nổi. Trong giai đoạn này, rõ ràng là các câu hỏi và các ý tưởng sẽ xuất hiện và chúng cần phải được bổ sung thêm vào mô

hình Use Case trước khi đến giai đoạn phê duyệt chung cuộc. Giai đoạn xác nhận cũng có thể được thực hiện trong thời kỳ thử nghiệm hệ thống, nhưng điểm yếu của phương thức làm này là nếu hệ thống không thỏa mãn những yêu cầu cụ thể của người sử dụng thì toàn bộ dự án rất có thể sẽ phải làm lại từ đầu.

Kiểm tra hệ thống là để đảm bảo nó hoạt động đúng như đặc tả. Điều này không thể được thực hiện trước khi đã có những thành phần của hệ thống được tạo ra. Chỉ sau đó người ta mới có thể thử xem hệ thống có hoạt động đúng như đặc tả mà người sử dụng đã đưa ra, rằng các Use Case thực hiện đúng theo như những lời đã miêu tả trong mô hình, rằng chúng hoạt động theo đúng phương thức đã được miêu tả trong văn bản miêu tả Use Case.

Chương 4 CHƯƠNG TRÌNH ỨNG DỤNG

4.1 Bài toán quản lý văn bản đến và văn bản đi

Hiện nay, Ngân hàng PG Bank có 28 chi nhánh trên cả nước. Việc gửi và nhận các thông báo, chứng từ, văn bản trao đổi được thực hiện thông qua mail, fax, chuyển phát rất dễ bị thất lạc và không kịp thời. Xuất phát từ thực tế đó xây dựng hệ thống quản lý văn bản.

- Mục tiêu của chương trình là nhằm nâng cao hiệu quả tác nghiệp giữa các phòng ban và chi nhánh trong ngân hàng trong việc luân chuyển các văn bản, thông báo,.. đến và đi bằng việc sử dụng kỹ thuật phân cụm dữ liệu trong hệ QTCDL Oracle.

Phát biểu bài toán.

Input: Các văn bản đến và đi

- *Văn bản đến:* là các hồ sơ, tài liệu (đơn hàng, văn bản, hồ sơ, tài liệu, báo cáo của các doanh nghiệp; đề án; giấy mời...), văn bản pháp qui do các cơ quan gửi đến Cty trực tiếp hoặc qua đường văn thư.

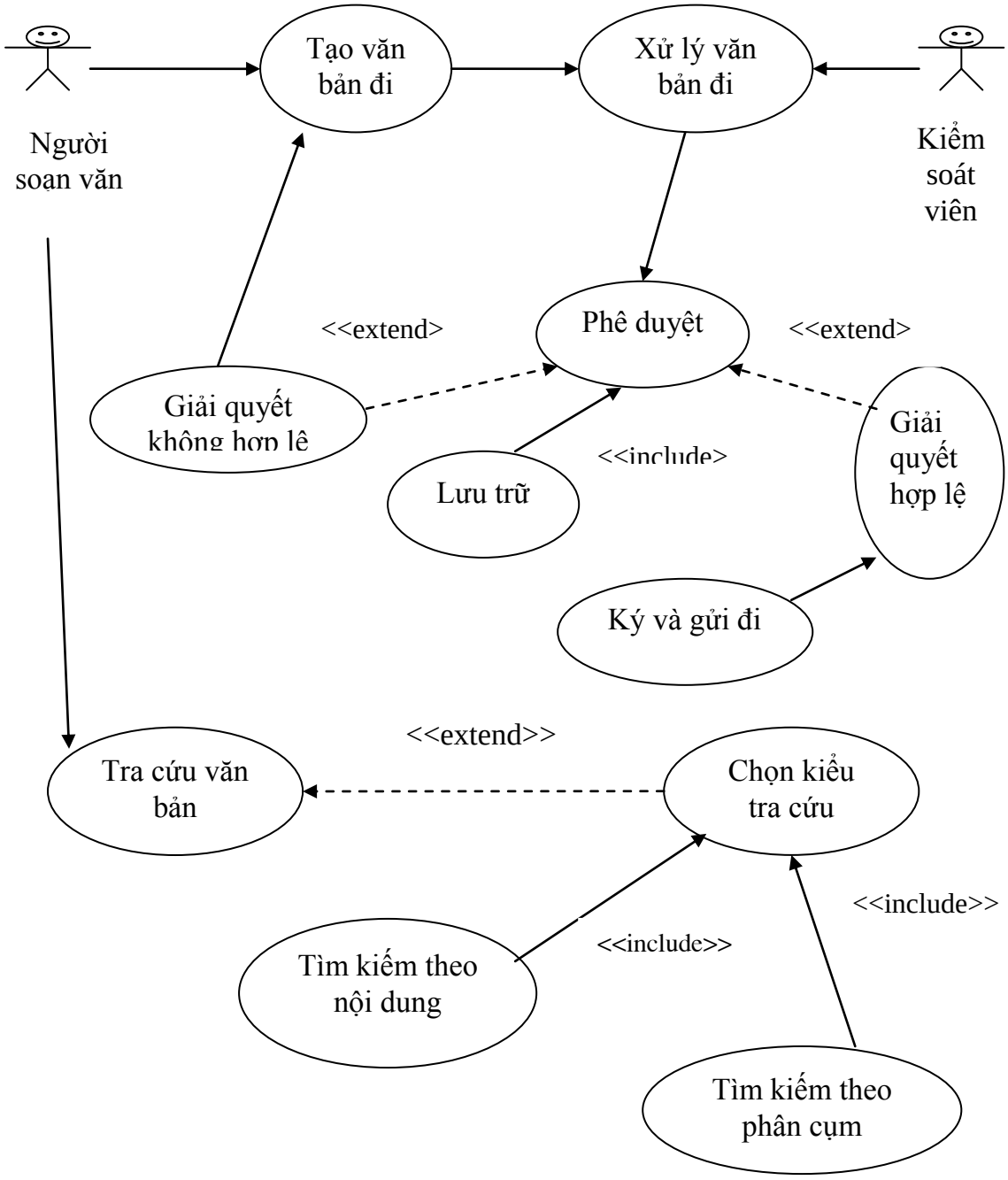
- *Văn bản đi:* là các hồ sơ, tài liệu, văn bản trả lời, quyết định, giấy báo, giấy mời... các cá nhân, doanh nghiệp...do Cục phát đi cho các đơn vị, cá nhân theo con đường trực tiếp hoặc qua đường văn thư.

Output: Hệ thống phải phân loại được các văn bản nhanh, chính xác, sử dụng thuật toán phân cụm để hỗ trợ cho việc tra cứu, tìm kiếm...

4.2 Mô hình usecase trong hệ thống quản lý văn bản đến và đi

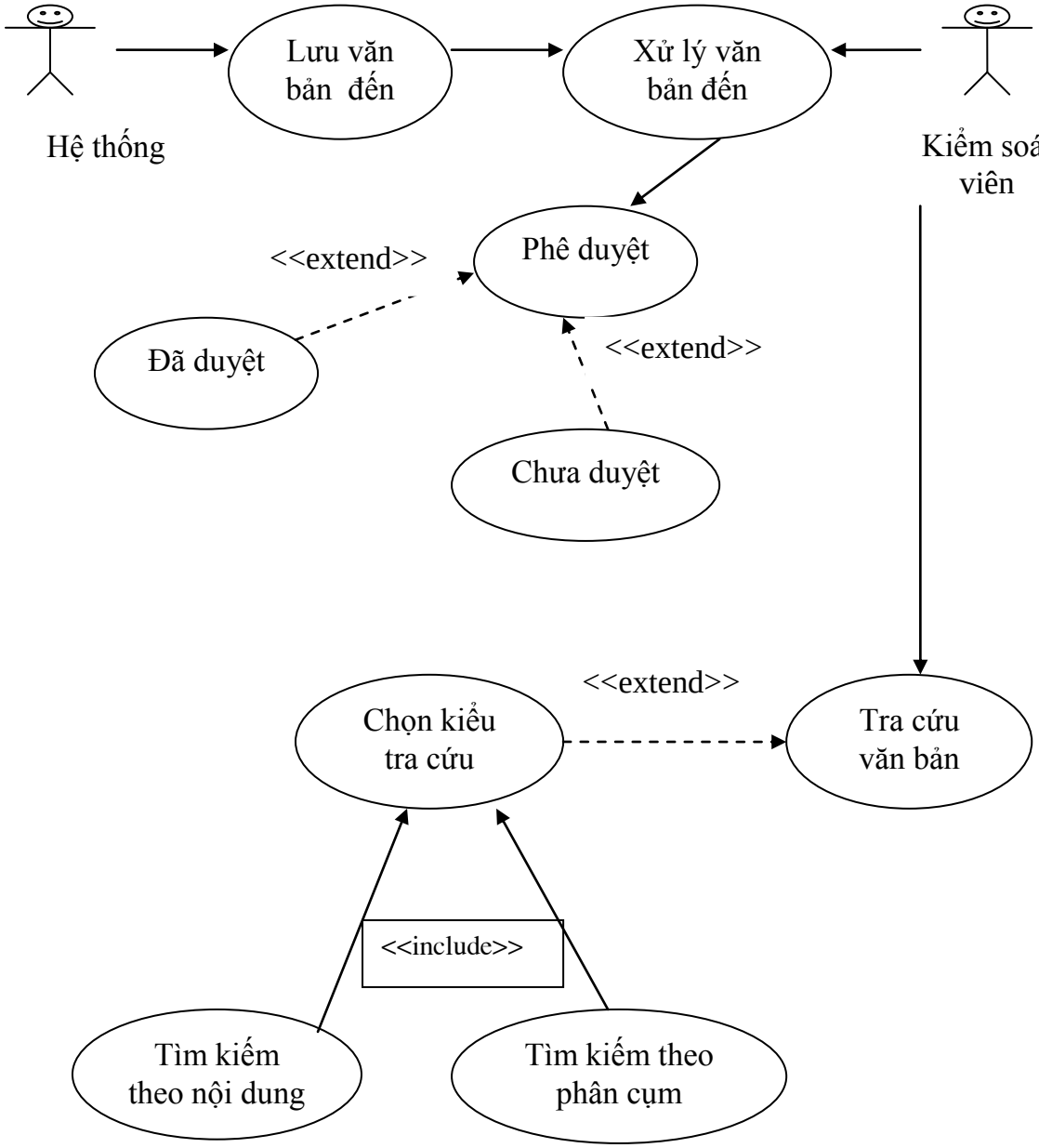
4.2.1 Quy trình tạo, gửi đi

Nhân viên tạo văn bản đăng nhập vào tạo văn bản. Hồ sơ được lưu dưới trạng thái input (marker). sau đó kiểm soát viên phê duyệt, ký (trưởng phòng, phó phòng) lúc đó hồ sơ mới được gửi đi.


Người soạn văn
Tạo văn bản đi
Xử lý văn bản đi
Kiểm soát viên
Phê duyệt
<<extend>
<<extend>
Giải quyết không hợp lệ
Lưu trữ
<<include>
Giải quyết hợp lệ
Ký và gửi đi
<<extend>>
Tra cứu văn bản
Chọn kiểu tra cứu
<<include>>
<<include>>
Tìm kiếm theo nội dung
Tìm kiếm theo phân cụm

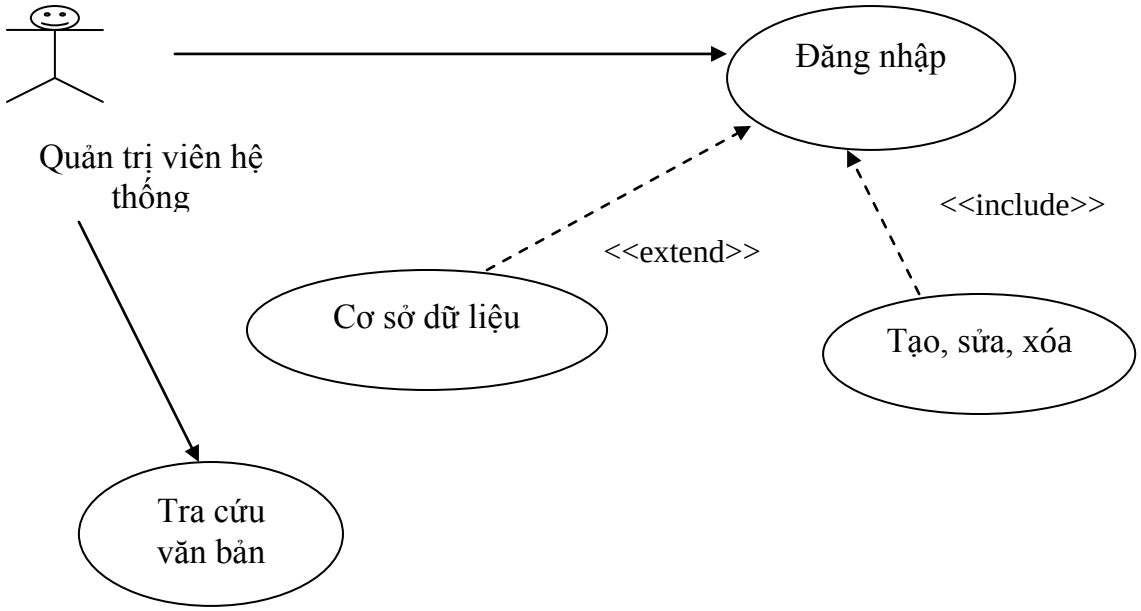
4.2.2 Quy trình nhận, đến

Văn bản đến được lưu tự động. Kiểm soát duyệt (Trưởng phòng, phó phòng) (checker). Sau đó văn bản đến được lưu vào danh mục những văn bản đã duyệt.



4.2.3 Quản trị viên hệ thống:

Tạo và phân quyền cho người dùng thuộc nhóm user, kiểm soát hay phê duyệt. quyền tạo mới, sửa đổi hay hủy hồ sơ về nhân viên, kiểm sát viên, phòng ban.



Danh sách usercase

1.Quy trình tạo, gửi đi

Tạo văn bản đi, xử lý văn bản đi , phê duyệt, lưu trữ,giải quyết hợp lệ, giải quyết không hợp lệ, ký và gửi đi, tra cứu.

2. Quy trình nhận, đến

Lưu văn bản đến, xử lý văn bản đến, phê duyệt, đã duyệt, chưa duyệt, tra cứu.

3.Quản trị

Đăng nhập, CSDL nhân viên, phòng ban, Tạo, sửa, xóa, tra cứu.

4.3 Đặc Tả User Case

Giới Thiệu:

Chương trình Quản lý văn bản được xây dựng nhằm các mục đích sau:

- Tinhọc hoá quá trình lưu trữ và xử lý văn bản của một doanh nghiệp.
- Giảm công sức, chi phí lưu trữ, tìm kiếm và xử l. công văn
- Tự động hoá các quá trình nhận- gửi công văn
- Hỗ trợ ban giám đốc theo dõi tình trạng xử lý văn bản của các phòng ban
- Lưutrữ văn bản an toàn, tìm kiếm dễ dàng

 Mọi quá trình thực hiện việc xử lý văn bản, công việc đều được hệ thống máy tính ghi nhận nên lãnh đạo sẽ có biện pháp thích hợp để điều chỉnh các hoạt động của các phòng ban chức năng.

2. Bảng Chú Giải:

Phần này nhằm chú giải cho các thuật ngữ đã dùng trong bài này.

- Các định nghĩa:
    - Nhân viên : Người mà sử dụng chương trình, soạn, nhận lệnh từ cấp trên.
    - Kiểm soát viên: Người phê duyệt, lưu trữ, ký, gửi văn bản đi, nhận văn bản đến.
    - Admin: Người quản lý chương trình, tạo mới, sửa, xóa nhân viên, kiểm soát viên (trưởng phòng, phó phòng), các phòng ban.

1. Use Case Model:

- Mô hình Use Case: Như trên.
- Tra Cứu văn bản :
    - Tóm tắc: Use Case này mô tả cách mà một người tra cứu văn bản thông qua hệ thống này.
    - Dòng sự kiện:

      + Dòng sự kiện chính: Use case này bắt đầu khi một người muốn tra cứu một văn bản.

        1. Tìm kiếm theo chủ đề. Hệ thống yêu cầu chọn kiểu tra cứu (có thể tra cứu theo tên, loại,ngày, thuộc bộ phận nào...,theo mã số.Khi các thông tin đã được nhập đầy đủ hệ thống sẽ thực hiện và cho kết quả.
        2. Tìm kiếm theo phân cụm: Ta cần nhập số cụm, mỗi cụm sẽ cho ta biết thông tin về một chủ đề nào đó, từ đó ta rút ra được những thông tin hữu ích hỗ trợ cho việc ra quyết định.

      + Dòng sự kiện khác: Không tìm thấy thông tin như yêu cầu thì thông báo không tìm thấy.
    - Các Yêu Cầu đặt biệt: Cần phải nhập số cụm, theo cảm tính hoặc kinh nghiệm.
    - Điều kiện tiên quyết: Trước tiên nhân viên phải chọn kiểu tra cứu

và nhập số cụm.

- Post-Conditions: Nếu Use Case thàng công thì sẽ cho kết quả tìm được,hoặc không thành công thì thông báo lỗi.
- Điểm mở rộng: Không có.

- Tạo văn bản đi :
  - Tóm tắt: Use case mô tả những hoạt động tạo văn bản như tạo, sửa, xóa, thêm mới..
  - Dòng sự kiện:

    + Dòng sự kiện chính:Use Case này sẽ được bắt đầu khi người soạn công văn tạo công văn.

    1. Người soạn nhập các thông tin văn bản cần thiết
    2. Gửi thông tin đến Server.

    + Dòng sự kiện khác: Thông tin gửi đi bị lỗi,hệ thống sẽ thông báo cho khách hàng để khách hàng thực hiện lại thao tác.
  - Post-Conditions: Nếu Use Case thành công,công văn sẽ được thêm vào hệ thống,ngược lại hệ thống không thay đổi.
  - Điểm mở rộng: Không có.
- Xử lý văn bản đi:
  - Tóm tắt: Use Case mô tả những xử lý văn bản đi như kiểm tra nội dung, hình thức, ký...
  - Dòng sự kiện:

    + Dòng sự kiện chính: Use Case sẽ được bắt đầu khi thông tin từ người tạo văn bản chuyển đến

    1. Kiểm soát viên checker nội dung và hình thức có đúng quy cách ko ?
    2. Đồng ý chuyển tới phê duyệt, ký và gửi đi.

    + Dòng sự kiện khác: Không đồng ý yêu cầu chỉnh sửa lại
  - Các Yêu Cầu đặt biệt: Không có.
  - Điều kiện tiên quyết: Có công văn tạo chuyển đến,chờ ký
  - Post-Conditions: Nếu Use Case thành công, văn bản sẽ được chuyển tới phê duyệt.
  - Điểm mở rộng: Không có.

- Phê duyệt:
    - Tóm tắc: Use Case mô tả cách thức văn bản đi được lãnh đạo phê duyệt.
    - Dòng sự kiện:

      + Dòng sự kiện chính: Use Case sẽ bắt đầu khi văn bản đã được kiểm tra tới

      1. lãnh đạo xem và phê duyệt gửi đi
      2. Lưu trữ

      + Dòng sự kiện khác: Không đồng ý yêu cầu làm lại.
    - Các Yêu Cầu đặt biệt: Không có.
    - Điều kiện tiên quyết: Phải có sự checker cua kiểm sát viên Post-Conditions: Use Case thành công,công văn được gửi đi và ký, lưu trữ
- Lưu văn bản đến :
    - Tóm tắc: Use case mô tả những hoạt động lưu văn bản đến.
    - Dòng sự kiện:

      + Dòng sự kiện chính:Use Case này sẽ được bắt đầu khi văn bản được gửi đến.

      1.Hệ thống lưu các văn bản theo từng mức độ như công văn khẩn, công văn nội bộ..

      2.Gửi thông tin đến bộ phận xử lý.
    - Post-Conditions: Nếu Use Case thành văn bản đến sẽ được lưu vào hệ thống,ngược lại hệ thống không thay đổi.
    - Điểm mở rộng: Không có.
- Xử lý văn bản đến:
    - Tóm tắc: Use Case mô tả những xử lý văn bản đến như kiểm tra nội dung, hình thức..
    - Dòng sự kiện:

      + Dòng sự kiện chính: Use Case sẽ được bắt đầu khi văn bản chuyển đến

      1.Kiểm soát viên checker

      2.Đồng ý chuyển tới phê duyệt (đã xem) chờ chỉ đạo triển khai

- Điều kiện tiên quyết: Có văn bản lưu
- Post-Conditions: Nếu Use Case thành công, văn bản sẽ được duyệt và trả lời.
- Điểm mở rộng: Không có.

- Phê duyệt:
  - Tóm tắt: Use Case mô tả cách thức văn bản đến được lãnh đạo phê duyệt.
  - Dòng sự kiện:

    + Dòng sự kiện chính: Use Case sẽ bắt đầu khi văn bản tới

      1.lãnh đạo xem và phê duyệt

      2. Lưu trữ

    + Dòng sự kiện khác: Không.
  - Các Yêu Cầu đặt biệt: Không có.
  - Điều kiện tiên quyết: Phải có sự checker cua kiểm sát viên Post-Conditions: Use Case thành công,công văn được xem và lưu trữ, trả lời
  - Điểm mở rộng: Không có.
- Đăng Nhập:
  - Tóm tắt: Use Case mô tả cách nhân viên đăng nhập vào hệ thống.
  - Dòng sự kiện:

    + Dòng sự kiện chính: Use Case sẽ bắt đầu khi nhân viên đăng nhập.

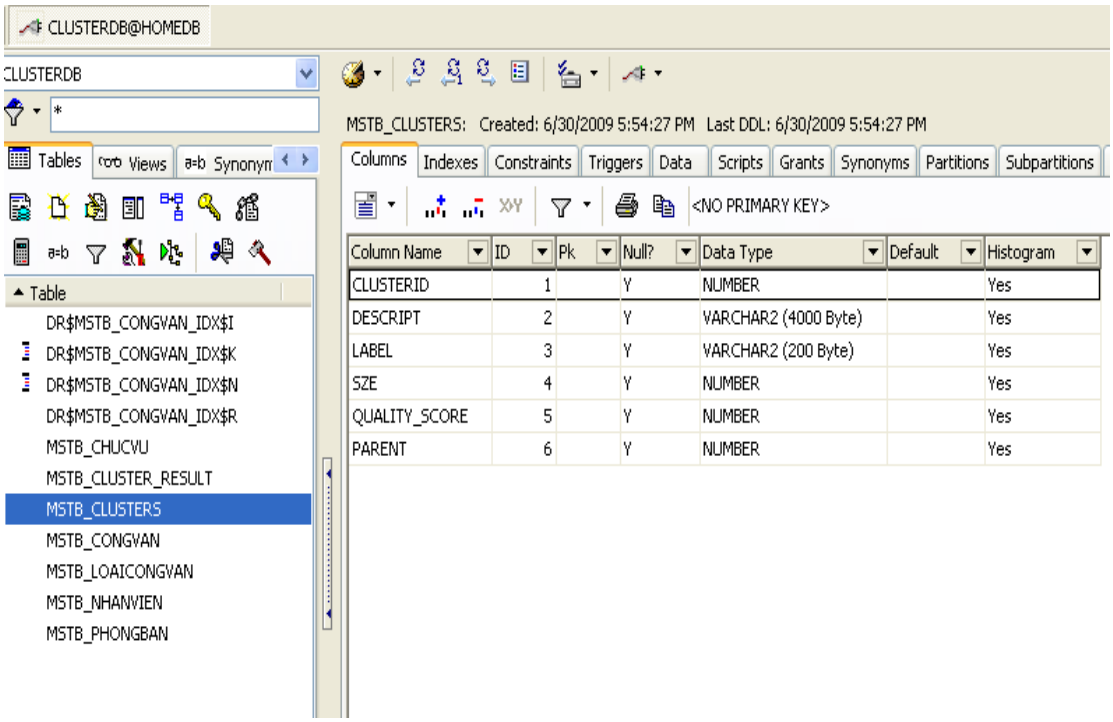
      1. Hệ thống yêu cầu nhân viên nhập Tên và Mật khẩu.
      2. Nhân viên nhập Tên và Mật khẩu
      3. Hệ thống kiểm chứng và cho nhân viên đăng nhập vào hệ thống.

    + Dòng sự kiện khác: Khi nhân viên nhập sai tên hoặc mật khẩu thì hệ thống sẽ thông báo lỗi và cho nhân viên chọn đăng nhập tiếp hay là kết thúc Use Case.
  - Các Yêu Cầu đặt biệt: Không có.
  - Điều kiện tiên quyết: Không có.
  - Post-Conditions: Nếu đăng nhập thành công thì nhân viên được

phép thao tác với những quyền của mình trong hệ thống,ngược lại hệ thống không thay đổi.

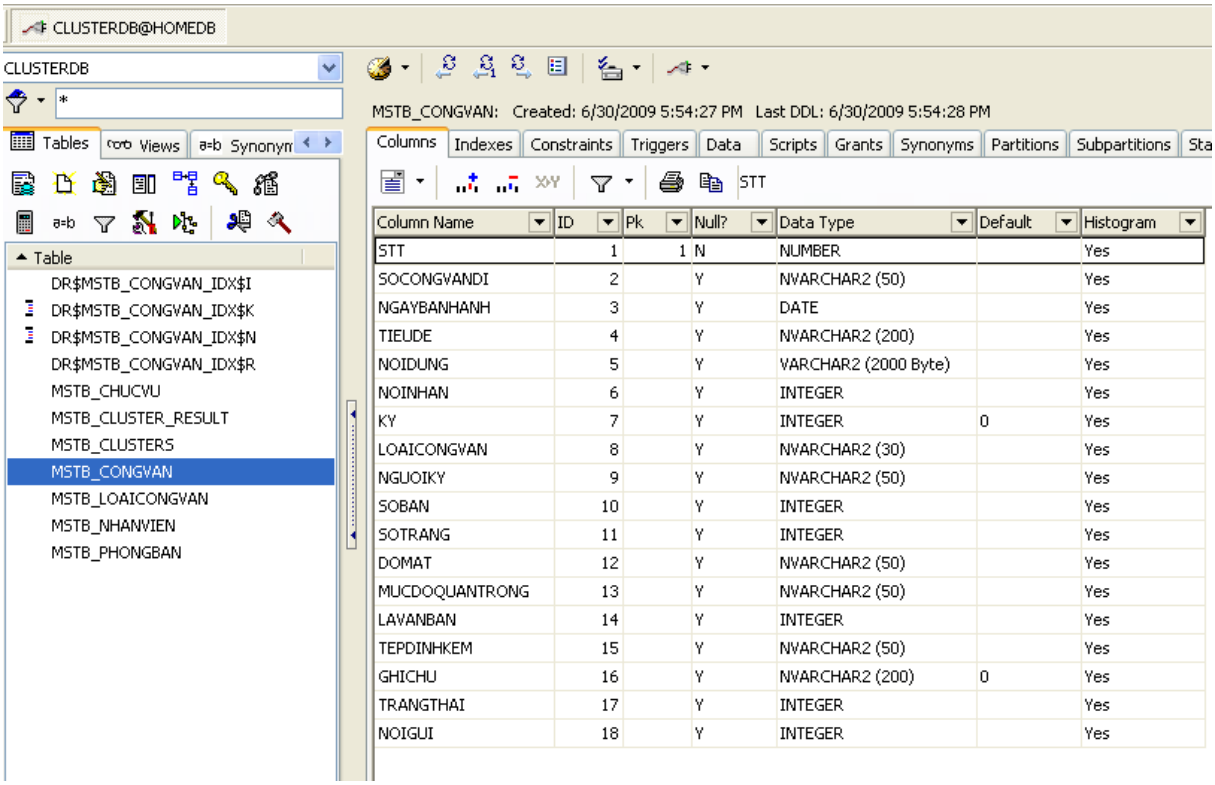
- Điểm mở rộng: Không có.

4.4 CSDL được tạo trong Oracle



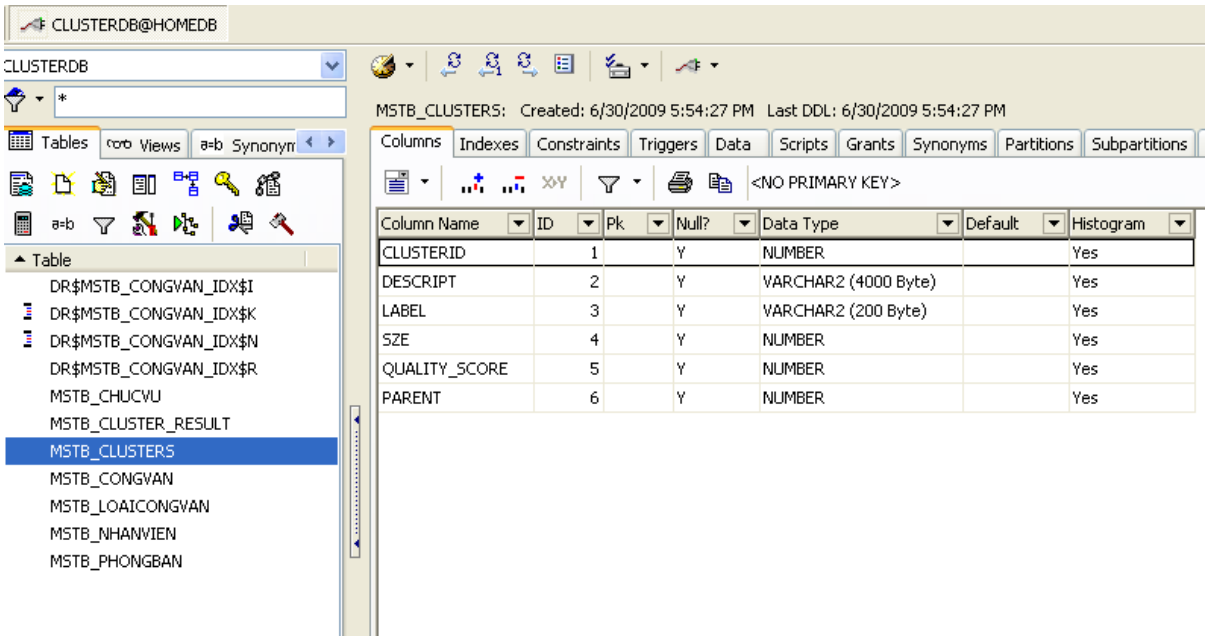
Column Name	ID	Pk	Null?	Data Type	Default	Histogram
CLUSTERID	1		Y	NUMBER		Yes
DESCRIPT	2		Y	VARCHAR2 (4000 Byte)		Yes
LABEL	3		Y	VARCHAR2 (200 Byte)		Yes
SZE	4		Y	NUMBER		Yes
QUALITY_SCORE	5		Y	NUMBER		Yes
PARENT	6		Y	NUMBER		Yes

4.5 Bảng MSTB_CÔNG VĂN

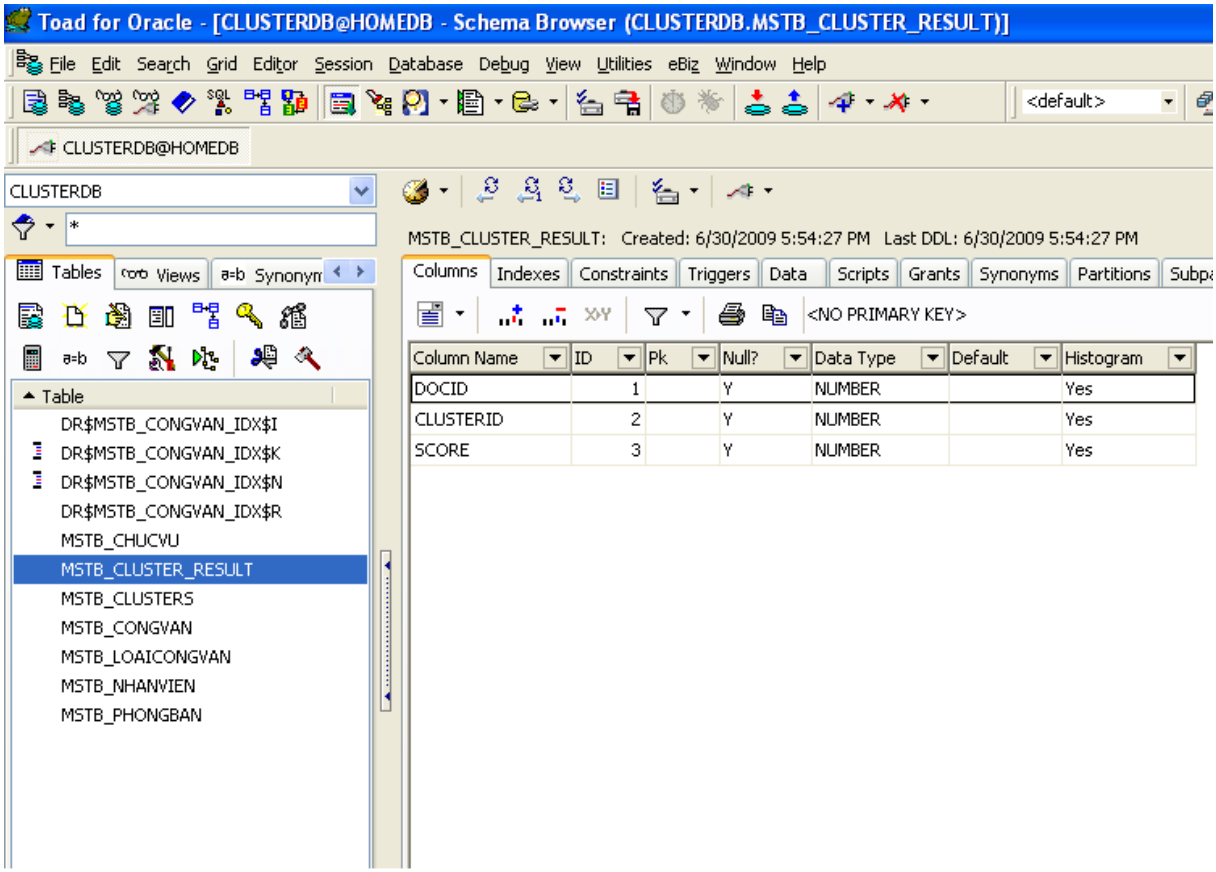


Column Name	ID	Pk	Null?	Data Type	Default	Histogram
STT	1	1	N	NUMBER		Yes
SOCONGVANDI	2		Y	NVARCHAR2 (50)		Yes
NGAYBANHANH	3		Y	DATE		Yes
TIEUDE	4		Y	NVARCHAR2 (200)		Yes
NOIDUNG	5		Y	VARCHAR2 (2000 Byte)		Yes
NOINHAN	6		Y	INTEGER		Yes
KY	7		Y	INTEGER	0	Yes
LOAICONGVAN	8		Y	NVARCHAR2 (30)		Yes
NGUOIKY	9		Y	NVARCHAR2 (50)		Yes
SOBAN	10		Y	INTEGER		Yes
SOTRANG	11		Y	INTEGER		Yes
DOMAT	12		Y	NVARCHAR2 (50)		Yes
MUCDOQUANTRONG	13		Y	NVARCHAR2 (50)		Yes
LAVANBAN	14		Y	INTEGER		Yes
TEPDINHKEM	15		Y	NVARCHAR2 (50)		Yes
GHICHU	16		Y	NVARCHAR2 (200)	0	Yes
TRANGTHAI	17		Y	INTEGER		Yes
NOIGUI	18		Y	INTEGER		Yes

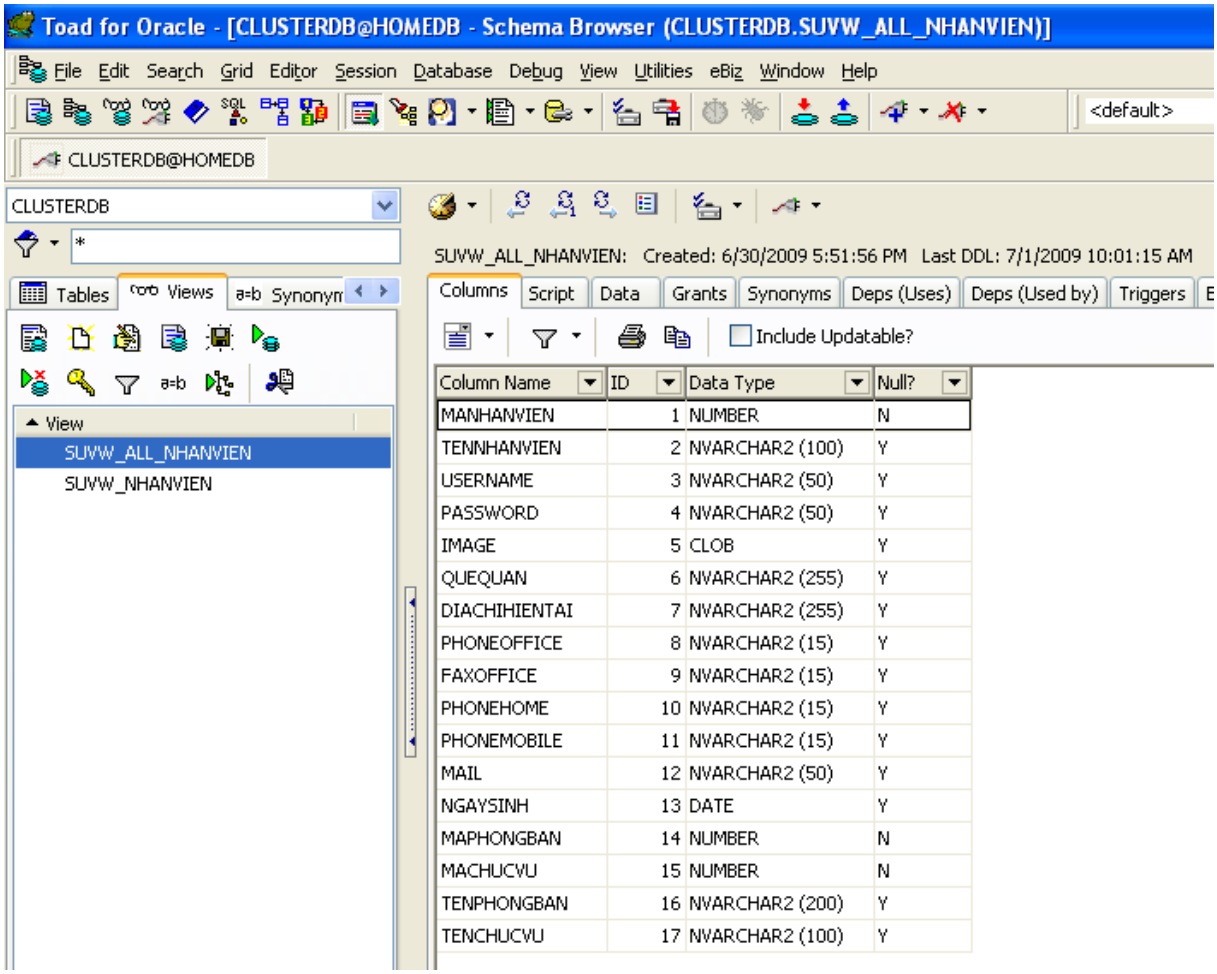
4.6 Bảng MSTB_CLUSTERS



4.7 Bảng MSTB_CLUSTER_RESULT

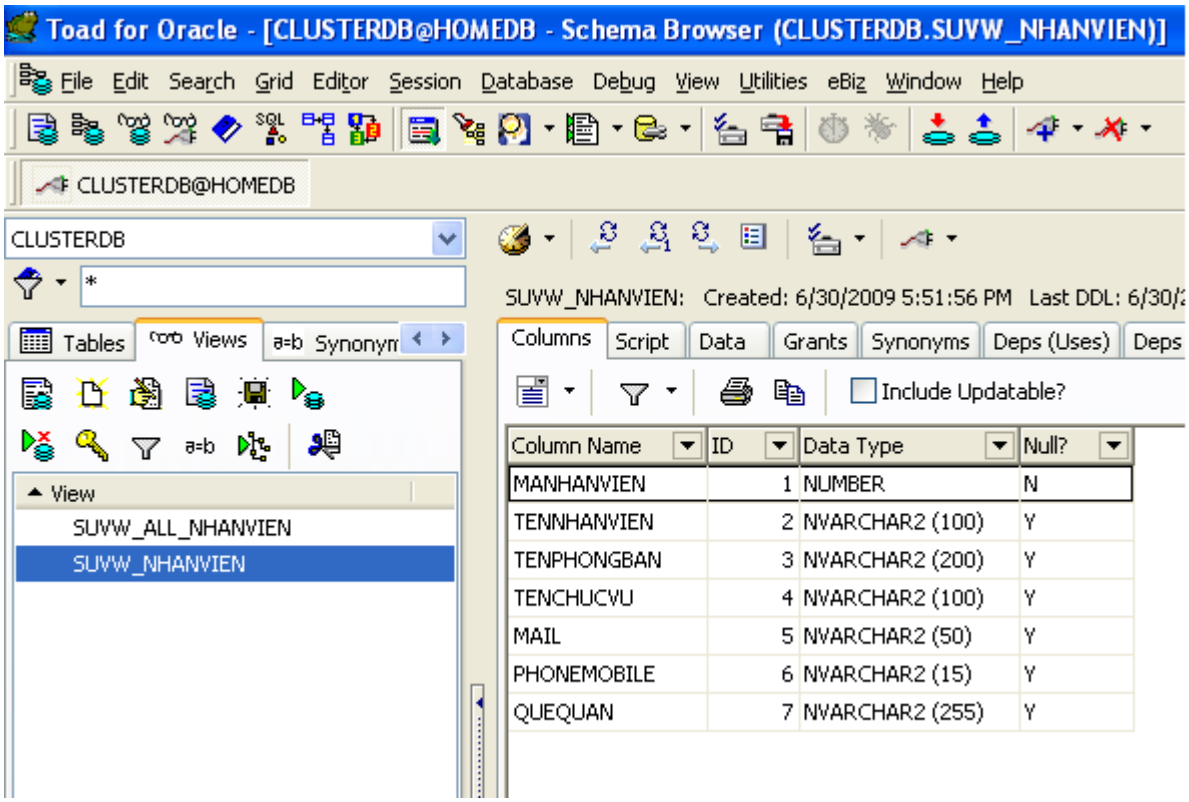


4.8 View tất cả nhân viên

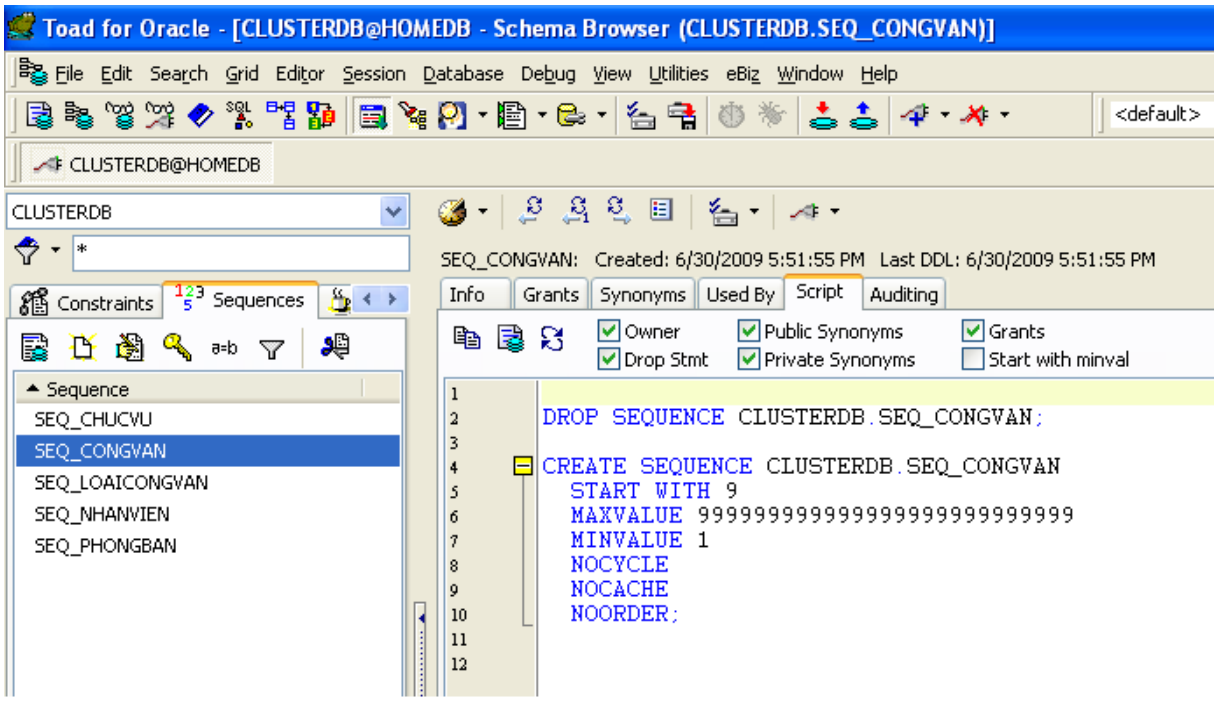


Column Name	ID	Data Type	Null?
MANHANVIEN	1	NUMBER	N
TENNHANVIEN	2	NVARCHAR2 (100)	Y
USERNAME	3	NVARCHAR2 (50)	Y
PASSWORD	4	NVARCHAR2 (50)	Y
IMAGE	5	CLOB	Y
QUEQUAN	6	NVARCHAR2 (255)	Y
DIACHIHIENTAI	7	NVARCHAR2 (255)	Y
PHONEOFFICE	8	NVARCHAR2 (15)	Y
FAXOFFICE	9	NVARCHAR2 (15)	Y
PHONEHOME	10	NVARCHAR2 (15)	Y
PHONEMOBILE	11	NVARCHAR2 (15)	Y
MAIL	12	NVARCHAR2 (50)	Y
NGAYSINH	13	DATE	Y
MAPHONGBAN	14	NUMBER	N
MACHUCVU	15	NUMBER	N
TENPHONGBAN	16	NVARCHAR2 (200)	Y
TENCHUCVU	17	NVARCHAR2 (100)	Y

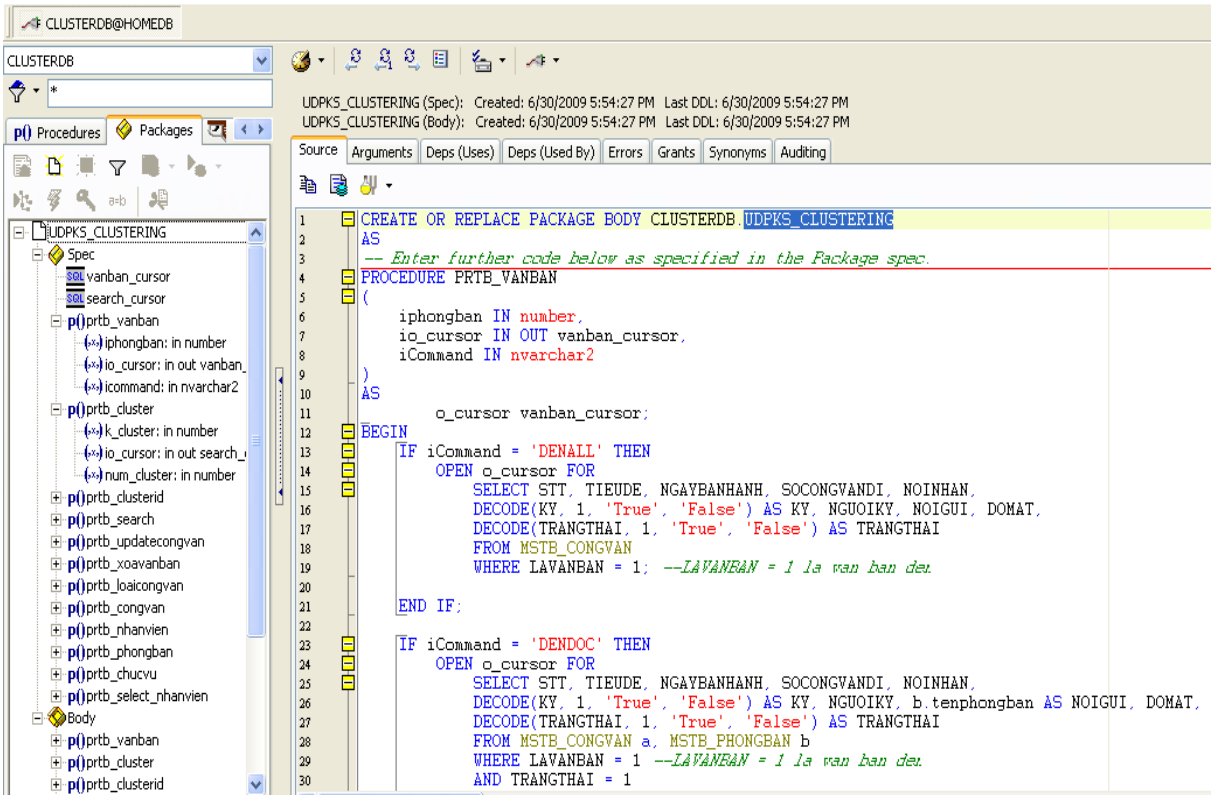
4.9 View nhân viên



4.10 Sequences



4.11 Trong Packages chứa các Procedures p()prtb_vanban,p()prtb_cluster...

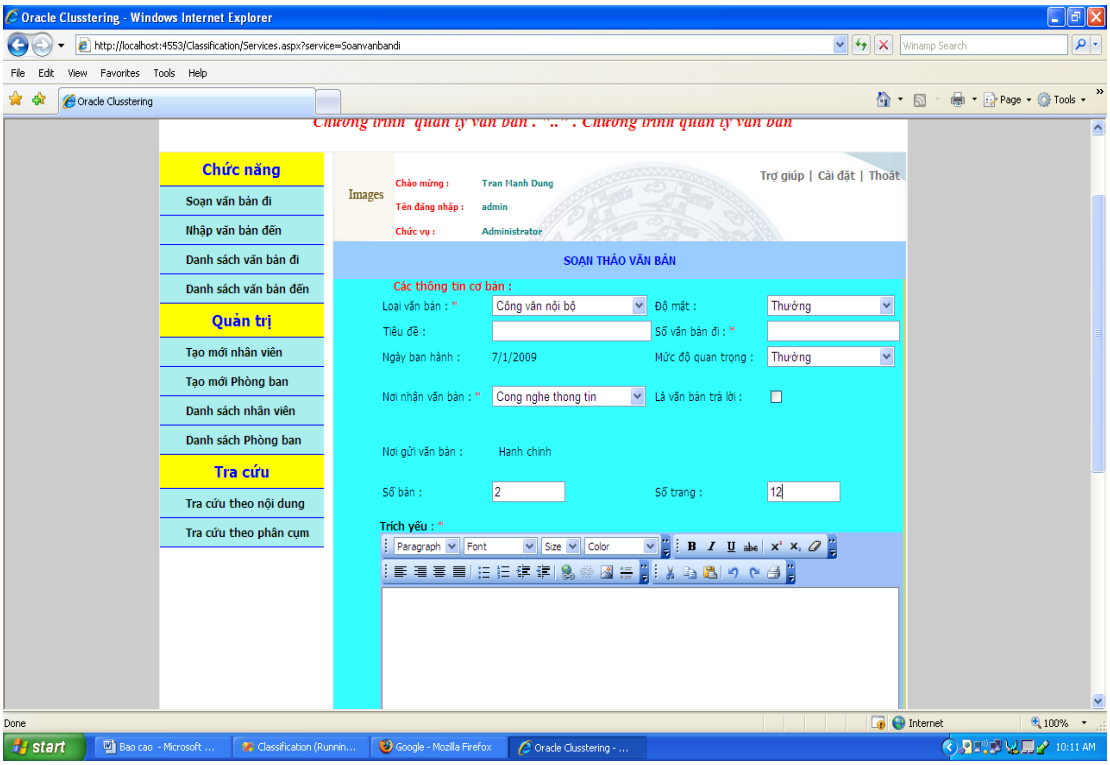


4.12 Giới thiệu chương trình ứng dụng

4.12.1 Trang Đăng nhập

4.12.2 Trang chủ

4.12.3 Trang Soạn văn bản



4.12.4 Trang Danh sách nhân viên

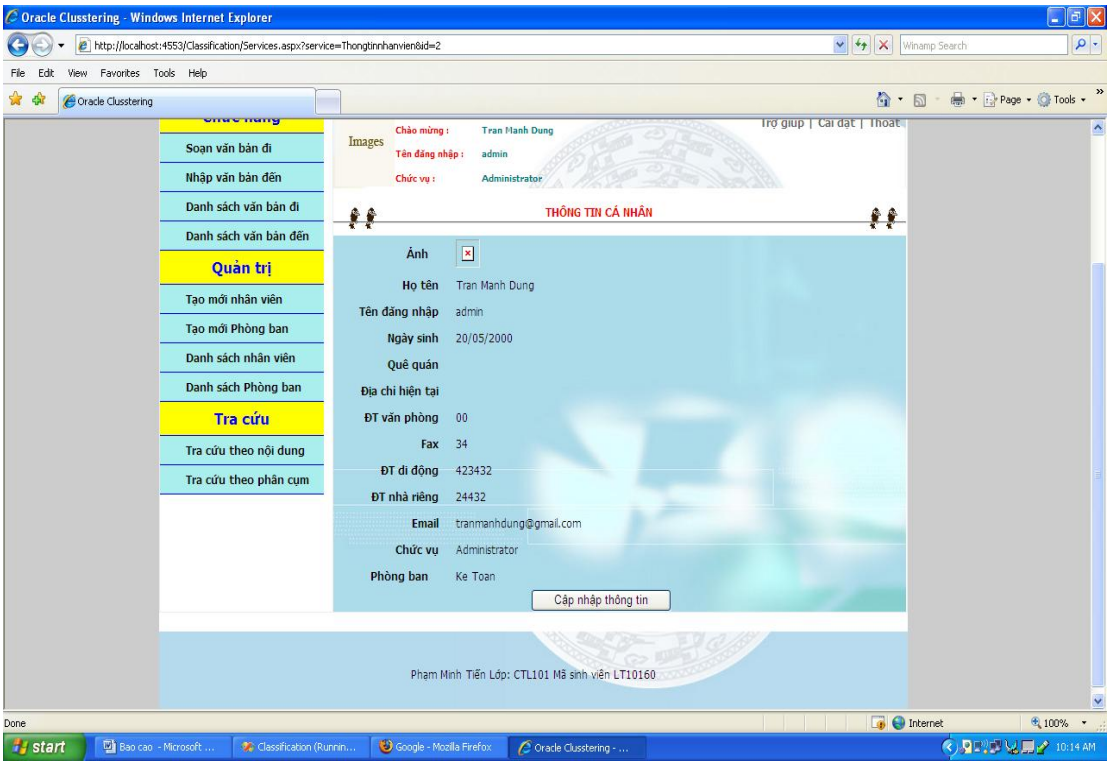
4.12.5 Trang tạo mới nhân viên

4.12.6 Trang danh sách phòng ban

4.12.7 Trang danh sach văn bản đến

4.12.8 Trang tạo mới phong ban

4.12.9 Trang thông tin cá nhân

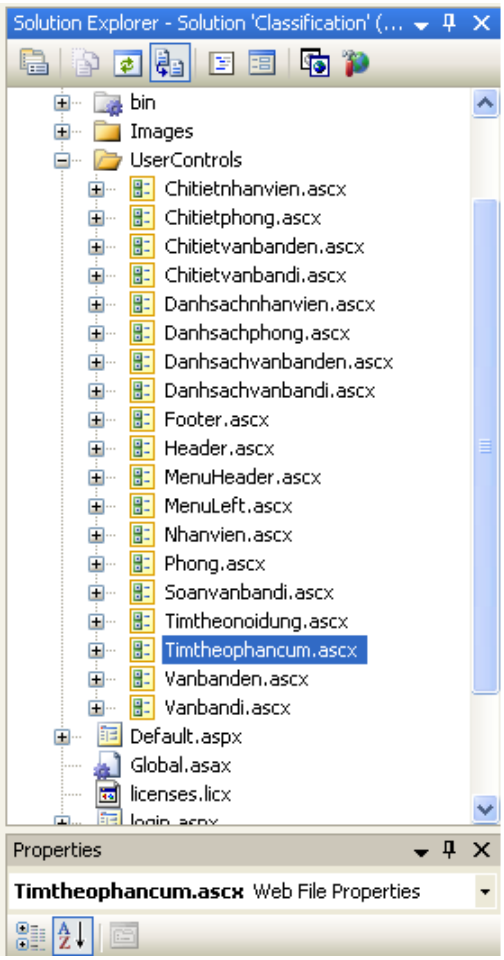


4.12.10 Trang tra cứu theo nội dung

4.12.11 Trang tra cứu theo phân cụm và kết quả chạy trương trình

4.13 Chương trình được thiết kế bởi Microsoft Visual Studio 2005

Danh mục cần thiết kế



4.14 Kết quả thực hiện chương trình

Chương trình thực hiện với bộ dữ liệu với 500 văn bản

Nhận xét: Đây là chương trình thực hiện phân cụm trên một bài toán cụ thể là Quản lý văn bản, qua đó kiểm nghiệm được kết quả của thuật toán phân cụm dữ liệu k_mean trên hệ quản trị cơ sở dữ liệu Oracle.

- Chương trình đã chạy với dữ liệu đầu vào là văn bản đến và đi khi phân cụm toàn bộ văn bản đến và đi với số cụm K chọn ngẫu nhiên hoặc theo kinh nghiệm của chuyên gia. khi chạy chương trình sẽ cho ra kết quả các cụm số được phân, mỗi cụm sẽ có những tiêu chí, nội dung tương đồng nhau, hỗ trợ cho quá trình tra cứu tìm ra những bộ số giống nhau.
- Ưu điểm: Chương trình có khả năng ứng dung thực tế cao, chạy trên csdl lớn, nhanh
- Tuy nhiên, vẫn còn những hạn chế nhất định như là số cụm K chỉ có thể chọn ngẫu nhiên hoặc theo kinh nghiệm của chuyên gia.

KẾT LUẬN

Trong đồ án này, em đã trình bày tổng quan và các nét đặc trưng nhất trong lĩnh vực Data Mining bao gồm các vấn đề cần khám phá tri thức, các hướng tiếp cận nghiên cứu tiêu biểu, trong đó PCDL là một phương pháp khám phá tri thức quan trọng trong Data Mining có nhiều ý nghĩa trong khoa học cũng như thực tiễn.

Đồ án này đã tìm hiểu được 1 số vấn đề trong việc phân cụm dữ liệu trong hệ quản trị csdl Oracle như các thủ tục, các gói dữ liệu tích hợp trong Oracle....Các cách gọi, sử dụng thuật toán trong phân cụm.....

Đồ án đã xây dựng được một chương trình nhỏ quản lý văn bản có ý nghĩa tương đối cao làm tiền đề cho việc phát triển những ứng dụng sau này.

Hướng phát triển tiếp theo:

Đồ án đã đề cập đến một số các phương pháp cũng như kỹ thuật áp dụng trong PCDL. Với tiền đề đó, trong thời gian sắp tới, tôi sẽ tiếp tục tìm hiểu các mô hình dữ liệu đặc thù, và lựa chọn một một kỹ thuật PCDL phù hợp nhằm xây dựng các ứng dụng đáp ứng các bài toán trong thực tiễn. Hướng nghiên cứu cụ thể như sau:

- Xây dựng và phát triển các kỹ thuật phân cụm cho lớp các dữ liệu Web, văn bản, hình ảnh.
- Kết hợp các kỹ thuật phân cụm với các các kỹ thuật mờ, mạng nơ ron đề giải quyết một số ứng dụng khác trong thực tế.

Do thời gian nghiên cứu và trình độ có hạn, báo cáo không tránh khỏi có những hạn chế và thiếu sót. Em xin được tiếp thu ý kiến sự đánh giá, chỉ bảo của các thầy giáo cũng như các bạn bè.

Chương 5 TÀI LIỆU THAM KHẢO

[1]. Nguyễn Thị Ngọc, Thuật toán phân cụm dữ liệu dựa trên mật độ, Đồ án tốt nghiệp, ĐHDL Hải Phòng, 2008.

[2]. Trần Thị Quỳnh, Phân cụm dữ liệu nửa giám sát và giải thuật di truyền, Đồ án tốt nghiệp, ĐHDL Hải Phòng, 2008.

[3]. Kluwer Academic Publishers, Holland, *Extensions To the k-means Algorithm for Clustering Large Data Sets With Categorical Value*

[4]. Periklis Andritsos, *Data Clusting Techniques*, Department of Computer Science, University Toronto, 2002.

[5]. Tài liệu được cung cấp bởi đơn vị thực tập Ngân hàng TMCP xăng dầu Petrolimex

Các Website:

[1] http://www.oravn.com/

[2] http://www.oracle.com/technology/index.html