

Bài báo nghiên cứu

MỘT SỐ KIẾN THỨC, KINH NGHIỆM THIẾT KẾ CƠ SỞ DỮ LIỆU CÓ YẾU TỐ THỜI GIAN

Nguyễn Hồng Ly

Chi cục Tiêu chuẩn Đo lường Chất lượng Đồng Tháp, Việt Nam

Tác giả liên hệ: Nguyễn Hồng Ly – Email: dongthap.hongly@gmail.com

Ngày nhận bài: 25-7-2022; ngày nhận bài sửa: 22-02-2023; ngày duyệt đăng: 23-02-2023

TÓM TẮT

Hiện nay, các tài liệu nghiên cứu ứng dụng về cơ sở dữ liệu (CSDL) có yếu tố thời gian rất ít, các bài báo được chia sẻ chủ yếu là nghiên cứu lý thuyết, nội dung không đi sâu vào giải quyết một ứng dụng cụ thể. Vì vậy, việc nghiên cứu và đề xuất các giải pháp chi tiết cụ thể để các cá nhân, tổ chức có thể vận dụng dễ dàng và hiệu quả trong việc xây dựng CSDL đảm bảo tính đầy đủ, chính xác của thông tin. Từ đó giúp cho việc khai thác sử dụng thông tin mang lại hiệu quả.

Trong bài báo này, tác giả chia sẻ chi tiết về một giải pháp nghiên cứu xây dựng CSDL của một Hệ thống hỗ trợ công tác báo cáo, thống kê số liệu theo các biểu mẫu sử dụng trong cơ quan hành chính. Giải pháp này chia sẻ một quy trình xây dựng CSDL với các bước thực hiện cụ thể, trong đó có sự phân tích một số lỗi thường gặp liên quan đến việc áp dụng “máy móc” các lý thuyết phương pháp thiết kế. Đồng thời, đề xuất các cách giải quyết phù hợp để nâng cao hiệu quả xử lý CSDL, đáp ứng các nhu cầu sử dụng khác nhau trong thực tiễn.

Từ khóa: cơ sở dữ liệu; thiết kế; yếu tố thời gian

1. Giới thiệu

Chúng ta biết rằng, tất cả thông tin về sự tồn tại của các sự vật, hiện tượng trong thế giới thực luôn gắn liền với các yếu tố thời gian cụ thể của quá khứ, hiện tại hay tương lai. Do đó, những thông tin của thực thể khi được lưu trữ trong CSDL cần có đầy đủ các yếu tố thời gian liên quan. Bởi vì, các yếu tố thời gian này sẽ ảnh hưởng rất lớn đến sự hoàn chỉnh và tính chính xác của thông tin.

Tuy nhiên thực tế hiện nay, nhiều hệ thống thông tin có CSDL được thiết kế theo kiểu đơn giản hóa, bỏ qua các yếu tố thời gian. Dẫn đến hậu quả làm cho CSDL mất đi tính đầy đủ về mặt thời gian, ảnh hưởng đến sự chính xác của những thông tin được phản ánh và tính thực tiễn của ứng dụng.

Chẳng hạn như trong một hệ thống thông tin báo cáo, nếu CSDL của hệ thống không được thiết lập các yếu tố thời gian liên quan, để theo dõi sự biến động dữ liệu của các đối

Cite this article as: Nguyen Hong Ly (2023). Knowledge and experience in temporal database design. *Ho Chi Minh City University of Education Journal of Science*, 20(2), 218-231.

tượng theo thời gian, mà chỉ quan tâm đến việc lưu trữ các giá trị hiện tại. Thì báo cáo về các đối tượng đó cũng sẽ chứa thông tin không đảm bảo được tính đầy đủ, chính xác. Kết quả là quyết định được đưa ra dựa trên các báo cáo đó có thể sẽ trở nên sai lệch.

Như vậy, làm thế nào để quản lý dữ liệu thay đổi theo thời gian? Giải pháp cho vấn đề này chính là xây dựng CSDL có yếu tố thời gian (sau đây gọi tắt là CSDL thời gian).

CSDL thời gian cho phép lưu trữ và truy nhập các loại dữ liệu thời gian. Thuật ngữ dữ liệu thời gian có thể hiểu là dữ liệu hay đối tượng được xác định với một yếu tố thời gian (nhãn thời gian) cụ thể (Jensen & Snodgrass, 1999). Nói cách khác, một đối tượng có thể có nhiều giá trị khác nhau tại những thời điểm khác nhau.

Điểm khác biệt giữa CSDL thời gian và CSDL phi thời gian đó là: CSDL thời gian hỗ trợ lưu trữ thông tin của một đối tượng liên quan đến thời gian trong quá khứ, hiện tại và tương lai (nếu có) của nó. Trong khi, CSDL phi thời gian, chỉ lưu trữ các dữ kiện được cho là đúng tại thời điểm hiện tại (Khi dữ liệu được cập nhật, chèn hoặc xóa thì nó được viết đè lên trạng thái dữ liệu cũ và tạo nên một trạng thái mới trong CSDL. Và khi đó trạng thái trước khi có thay đổi xảy ra sẽ không còn nữa.

Yếu tố thời gian được đưa ra với nhiều ngữ nghĩa khác nhau phụ thuộc vào loại thời gian mà hệ thống tương ứng hỗ trợ. Có hai loại thời gian thường được sử dụng là thời gian giao tác (Transaction Time) và thời gian hợp lệ (Valid Time), cụ thể:

- Thời gian giao tác là thời gian lúc thực thể/sự kiện được lưu trữ trong CSDL. Nó được hệ thống ghi nhận và người dùng không thể can thiệp thay đổi.
- Thời gian hợp lệ của một sự kiện là thời gian mà sự kiện đó được xem là đúng trong thực tế. Nó được thiết lập bởi người dùng và có thể thay đổi.

Trong phần tiếp theo tác giả trình bày quy trình với 5 bước thực hiện để xây dựng CSDL của một Hệ thống thông tin hỗ trợ công tác báo cáo, thống kê số liệu theo các biểu mẫu sử dụng trong các cơ quan hành chính nhà nước (sau đây gọi tắt là Hệ thống BCTK) và kết quả thiết kế một số chức năng hỗ trợ truy vấn lịch sử dữ liệu của các thực thể có sự thay đổi theo thời gian trên hệ thống này.

2. Nội dung

a) Quy trình xây dựng CSDL thời gian của Hệ thống BCTK

Việc xây dựng CSDL thời gian của Hệ thống BCTK được thực hiện với 5 bước như sau:

Bước 1. Xác định mục đích của CSDL Hệ thống BCTK

Về cơ bản chức năng chính của Hệ thống BCTK là cung cấp các biểu mẫu để thu thập dữ liệu, xử lý, lưu trữ và kết xuất các biểu báo cáo thống kê. Do đó, mục đích của CSDL của Hệ thống BCTK được xác định là một CSDL có khả năng lưu trữ đầy đủ các dữ liệu cần thiết theo thời gian, để thuận lợi trong việc thực hiện những chức năng của Hệ thống BCTK, chẳng hạn như: dữ liệu có thể dễ dàng cập nhật, thông tin có thể truy xuất nhanh chóng, đảm bảo tính đầy đủ, chính xác nhằm phục vụ hiệu quả cho công tác quản lý và điều hành.

Bước 2. Thiết kế CSDL mức khái niệm

Công tác báo cáo thống kê trong các cơ quan hành chính được thực hiện theo sự phân công, phân cấp. Theo đó, mỗi **đơn vị** có trách nhiệm thực hiện các **chế độ báo cáo** (còn được gọi là **loại báo cáo**) theo định kì hoặc đột xuất liên quan đến lĩnh vực, phạm vi quản lí.

Việc thực hiện chế độ báo cáo thống kê do các **người dùng** thuộc mỗi đơn vị thực hiện, trên cơ sở **vai trò** chức trách được giao, để thực hiện các hoạt động nghiệp vụ báo cáo thống kê theo sự phân công (**ủy quyền**) như: thiết kế biểu mẫu, nhập liệu, kiểm tra, kí duyệt, ban hành.

Kết quả thực hiện đối với mỗi loại báo cáo thống kê, bao gồm: các biểu dữ liệu báo cáo thống kê tương ứng với mỗi **kì báo cáo** theo các **biểu mẫu** được quy định.

Đối với các biểu mẫu của mỗi loại báo cáo, có thể khác nhau về số **bảng biểu**, số hàng **chỉ tiêu**, số **cột thuộc tính**.

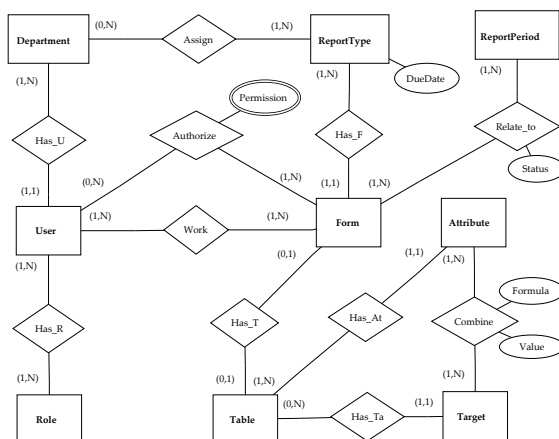
Biểu 04/KHCN-NV		NHIỆM VỤ KHOA HỌC VÀ CÔNG NGHỆ			- Đơn vị báo cáo:		
Ban hành kèm theo Thông tư số 15/2018/TT-BKHCN ngày 15 tháng 11 năm 2018		(Từ ngày đến ngày)				
Ngày nhận báo cáo: Ngày 15/02 năm sau					- Đơn vị nhận báo cáo:		
						
					Đơn vị tính: Nhiệm vụ		
					Tình trạng tiến hành		
					Số đang tiến hành		
					Số phê duyệt mới trong năm		
					Số chuyển tiếp từ năm trước		
					Số được nghiệm thu		
					Số đã đưa vào ứng dụng		
A	B	1	2	3	4	5	6
1. Tổng số nhiệm vụ KH&CN							
Trong đó: - Số đề tài/đề án KH&CN							
- Số dự án KH&CN							
2. Chia theo cấp quản lý							
- Cấp quốc gia							
- Cấp bộ							
- Cấp tỉnh							
- Cấp cơ sở							
3. Chia theo lĩnh vực nghiên cứu							
- Khoa học tự nhiên							
- Khoa học kỹ thuật và công nghệ							
- Khoa học y, dược							
- Khoa học nông nghiệp							
- Khoa học xã hội							
- Khoa học nhân văn							
					Đồng Tháp, ngày ... tháng ... năm 202...		
Người lập biểu		Người kiểm tra biểu			Thủ trưởng đơn vị		

Hình 1. Biểu mẫu báo cáo thống kê nhiệm vụ khoa học và công nghệ

Sau khi phân tích nghiệp vụ hoạt động báo cáo thống kê và các thành phần dữ liệu trong các biểu mẫu, có thể xác định các tập thực thể cần có trong CSDL của một Hệ thống BCTK, như sau:

STT	Tên tập thực thể	Giải thích
1	Department	Đơn vị
2	User	Người dùng
3	Role	Vai trò người dùng
4	ReportType	Loại báo cáo
5	ReportPeriod	Kì báo cáo
6	Table	Bảng biểu
7	Form	Biểu mẫu
8	Target	Chỉ tiêu
9	Attribute	Thuộc tính

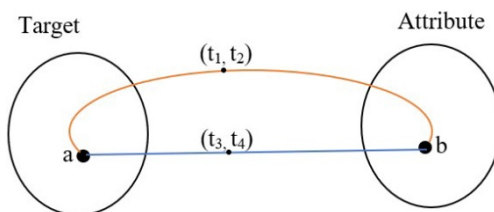
Xác định mối quan hệ trong thế giới thực giữa các thực thể trên ta có thể vẽ được mô hình thực thể kết hợp (ER) của CSDL Hệ thống BCTK như Hình 2:



Hình 2. Mô hình ER của CSDL hệ thống báo cáo thống kê

Sử dụng biểu đồ venn để biểu diễn các tập thực thể và các mối quan hệ của mô hình ER trên, ta có thể thấy mô hình này không phản ánh tốt khía cạnh thời gian trong thực tế của một số dữ liệu cần lưu trữ, đơn cử như vấn đề không thể biểu diễn được các giá trị của những cặp chỉ tiêu, thuộc tính của mối quan hệ kết hợp (Combine) theo thời gian.

Ví dụ: Trong thực tế chúng ta cần quản lý lịch sử các giá trị đã được lưu trữ của cặp chỉ tiêu, thuộc tính (a, b) của mối quan hệ kết hợp hai thực thể này, tương ứng với các khoảng thời gian (t_1, t_2) , (t_3, t_4) . Thông qua biểu đồ venn ta có thể biểu diễn các bộ giá trị của cặp chỉ tiêu, thuộc tính trên như Hình 3:



Hình 3. Biểu diễn các giá trị cặp chỉ tiêu, thuộc tính theo thời gian

Tuy nhiên, việc biểu diễn một mối quan hệ giữa hai thực thể nhiều hơn một đường nối, như biểu đồ venn trên (Hình 3) theo lý thuyết của mô hình ER là “không cho phép”.

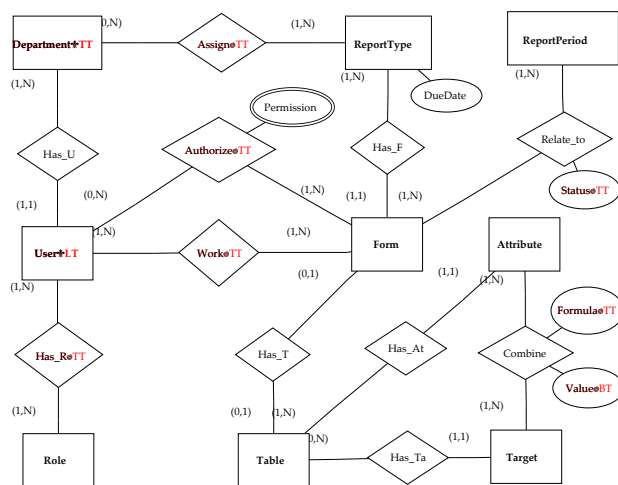
Ngoài ra, các thực thể Department (Đơn vị), User (Người dùng); các mối quan hệ Assign (Phân công), Relate_to (Liên quan), Authorize (Ủy quyền), Work (Làm việc), Has_R (Có vai trò), Has_F (Có biểu mẫu); các thuộc tính DueDate (Hạn báo cáo), Formula (Công thức), Value (Giá trị) cũng cần phải quản lý các dữ liệu thay đổi theo thời gian hoặc theo dõi lịch sử tương tác của người dùng.

Tuy nhiên, những yêu cầu đó khó có thể thực hiện được theo lý thuyết mà mô hình ER đưa ra, hay nói cách khác là mô hình ER không đảm bảo giải quyết đầy đủ vấn đề liên quan đến yếu tố thời gian của các thực thể, thuộc tính và mối quan hệ.

Hiện nay, có nhiều mô hình được đề xuất cho việc thiết kế CSDL thời gian mức khái niệm như: TimeER, TempEER, TempRT, TERC+... Trong đó, mô hình TimeER do (Gregersen & Jensen, 1999) xây dựng như là một mở rộng của mô hình ER với sự hỗ trợ khá đầy đủ các yếu tố thời gian so với các mô hình khác. Đây cũng chính là mô hình được tôi lựa chọn để thiết kế CSDL mức khái niệm của Hệ thống BCTK.

“Mô hình TimeER phát triển dựa trên mô hình ER. Mô hình này cho phép hỗ trợ các loại thời gian sau: thời gian sống của một thực thể, thời gian hợp lệ của một sự kiện và thời gian giao tác của một thực thể hoặc một sự kiện. Mô hình này quy ước rằng, đối với các thực thể, hệ thống chỉ có thể hỗ trợ thời gian sống (LifeSpan, kí hiệu là LS), hoặc thời gian giao tác (Transaction Time, kí hiệu là TT), hoặc cả hai loại thời gian này (kí hiệu là LT). Còn đối với các thuộc tính, hệ thống chỉ cho phép hỗ trợ thời gian hợp lệ (Valid Time, kí hiệu là VT), hoặc thời gian giao tác (TT) hoặc cả hai loại thời gian này (BiTemporal, kí hiệu là BT). Ngoài ra, do một mối quan hệ có thể xem là một tập thực thể hoặc một thuộc tính, nhờ vậy mà người thiết kế có thể xác định các yếu tố thời gian hỗ trợ cho mối quan hệ đó nếu cần”.

Dựa vào lí thuyết mô hình TimeER, mô hình CSDL mức khái niệm của Hệ thống BCTK có thể được hoàn thiện bằng việc bổ sung những loại thời gian liên quan, cần thiết cho sự quản lí đối với các thực thể, thuộc tính và mối quan hệ trong mô hình ER đã thiết kế ở trên, để được mô hình TimeER như Hình 4:



Hình 4. Mô hình CSDL Hệ thống Báo cáo thống kê mức khái niệm

Bước 3. Thiết kế CSDL thời gian mức Logic

Để chuyển đổi từ mô hình TimeER sang mô hình quan hệ, theo nghiên cứu (Hoang & Nguyen, 2012) đã xây dựng một phương pháp tổng quát dựa trên ý tưởng xây dựng các loại quan hệ (**Quan hệ chính** $R(E)$: lưu định danh các thực thể của kiểu thực thể E , kèm thông tin của các thuộc tính đơn trị; **Quan hệ thời gian** (TR): lưu trữ dữ liệu thời gian liên quan đến những thay đổi của các thực thể và các sự kiện trong một hệ thống; **Các quan hệ khác**: lưu thông tin về mối quan hệ giữa các kiểu thực thể và các dữ liệu đa trị của các thuộc tính. Các quan hệ chính đóng vai trò “trung tâm”).

Phương pháp trên đã giải quyết tương đối toàn diện các vấn đề liên quan đến việc chuyển đổi mô hình TimeER sang mô hình quan hệ.

Tuy nhiên, hiện nay các hệ quản trị CSDL được sử dụng phổ biến như SQL Server, MariaDB... đã có hỗ trợ bảng lưu trữ lịch sử dữ liệu kèm theo bảng lưu dữ liệu hiện thời, do đó trong trường hợp chuyển đổi một thực thể có yếu tố thời gian sang lược đồ quan hệ, thay vì phải xây dựng thêm lược đồ quan hệ thời gian tách khỏi lược đồ quan hệ chính như Phương pháp do (Hoang & Nguyen, 2012) đề xuất, thì chỉ cần xây dựng một lược đồ quan hệ có các thuộc tính của loại thời gian liên quan.

Theo đó, mô hình mức khái niệm (TimeER) của Hệ thống BCTK sẽ được chuyển đổi sang mô hình quan hệ, gồm các lược đồ quan hệ như sau:

- **Departments** (ID, DeptName, TTs_Dept, TTe_Dept).
- **Users** (UserName, Password, FullName, DeptID, LSs_User, LSe_User, TTs_User, TTe_User)
- **Roles** (ID, RoleName).
- **RoleOfUser** (UserID, RoleID, TTs_RU, TTe_RU).
- **ReportTypes** (ReportTypeID, ReportTypeName).
- **ReportingDeadlines** (ReportTypeID, DueDate, TTs_RpDeadline, TTe_RpDeadline).
- **ReportTypesToDepartments**(ReportTypeID, DeptID, TTs_RTD, TTe_RTD).
- **ReportPeriods** (ID, PeriodName, DateStart, DateEnd).
- **Forms** (ID, FormName, ReportTypeID).
- **Relate_PeriodForm** (FormID, PeriodID).
- **FormStatus** (ID, Status, TTs_Status, TTe_Status).
- **Work_UserForm**(UserID, FormID, Action, TTs_Work, TTe_Work).
- **Authorizations** (FormID, UserID, Permission, TTs_Authorization, TTe_Authorization).
- **Tables** (ID, TableName, FormID).
- **Attributes** (ID, AttributeName, DataTye, ParentID, TableID).
- **Targets** (ID, TargetName, DataType, IsParent, TableID).
- **AttributeTargetPairs** (TargetID, AttributeID).
- **Formulas_AT** (ID, Formula, TTs_Formula, TTe_Formula).
- **Values_AT** (ID, Value, VTs_Value, VTe_Value, TTs_Value, TTe_Value).

Trong quy trình thiết kế CSDL, thông thường sau khi chuyển đổi từ mô hình mức khái niệm sang mô hình quan hệ, sẽ tiến hành bước chuẩn hóa CSDL. Tuy nhiên, một lỗi chúng ta rất hay mắc phải là áp dụng các quy tắc chuẩn hóa mà không cần suy nghĩ về bản chất của ứng dụng, để rồi sau đó gặp các vấn đề về hiệu suất và phải tùy chỉnh.

Cho nên trước khi thực hiện công việc chuẩn hóa, chúng ta cần phải xác định bản chất của ứng dụng đang thiết kế là gì, là loại ứng dụng giao dịch (Transactional) hay ứng dụng phân tích (Analytical)?

Trong loại ứng dụng Transactional, người dùng cuối thường xuyên thực hiện các thao tác thêm, cập nhật, xóa các bản ghi. CSDL cho hệ thống này được gọi là OLTP.

Còn với ứng dụng Analytical, người dùng cuối thường quan tâm nhiều hơn đến phân tích, báo cáo, dự báo... Những loại CSDL này có số lần chèn và cập nhật ít hơn. Mục đích

chính ở đây là tìm nạp và phân tích dữ liệu nhanh nhất có thể. CSDL cho hệ thống này được gọi là OLAP.

Từ đó, nếu chúng ta thấy rằng việc chèn, cập nhật và xóa được thực hiện thường xuyên trên ứng dụng thì hãy thiết kế bảng chuẩn hóa (normalization), nếu không thì tạo cấu trúc CSDL không chuẩn hóa phẳng (denormalized).

Hệ thống BCTK là một ứng dụng phân tích dữ liệu, nhằm phục vụ cho việc tổng hợp, phân tích và báo cáo thông tin. Tuy nhiên, việc chuẩn hóa CSDL của hệ thống cần được xem xét, đánh giá cẩn thận để đảm bảo tính nhất quán và hiệu suất của hệ thống.

Áp dụng các quy tắc chuẩn hóa CSDL trong Hệ thống BCTK cho thấy lược đồ quan hệ của CSDL đã được thiết kế ở mức chuẩn hóa 3NF (Third Normal Form) – đây là mức chuẩn hóa phổ biến và phù hợp với nhiều hệ thống thông tin quản lí.

Việc chuẩn hóa CSDL giúp tránh được các vấn đề như sự lặp lại dữ liệu hoặc mất mát dữ liệu. Tuy nhiên, nếu không cần thiết, không nên chuẩn hóa CSDL ở mức cao hơn 3NF để tránh ảnh hưởng đến hiệu suất của hệ thống.

Bước 4. Thiết kế CSDL thời gian mức Vật lí

Trong thiết kế CSDL mức vật lí, các lược đồ quan hệ (LĐQH) sẽ được chuyển thành mô hình vật lí, tức là tập hợp các bảng. Theo quy tắc chuyển đổi, mỗi LĐQH và các thuộc tính của nó sẽ được chuyển thành bảng có các cột tương ứng và liên kết giữa các LĐQH trở thành ràng buộc giữa các bảng. Khóa chính của LĐQH sẽ chuyển thành khóa chính của bảng, và các LĐQH có yếu tố thời gian, khóa chính được đề xuất là khóa kết hợp giữa thuộc tính khóa của quan hệ chính và các thuộc tính thời gian.

Tuy nhiên, trong thực tế, ràng buộc khóa chính như vậy về lí thuyết rất chặt chẽ nhưng ít được sử dụng do hạn chế hiệu suất truy vấn. Vì vậy, để đơn giản hóa việc cài đặt và tối ưu tốc độ truy vấn ở một số bảng của CSDL, chúng ta có thể bổ sung thêm thuộc tính khóa chính ID kiểu INT và thiết lập tự động tăng, còn các thuộc tính khóa kết hợp thay vì được chọn làm khóa chính sẽ được đặt giá trị ràng buộc duy nhất (unique).

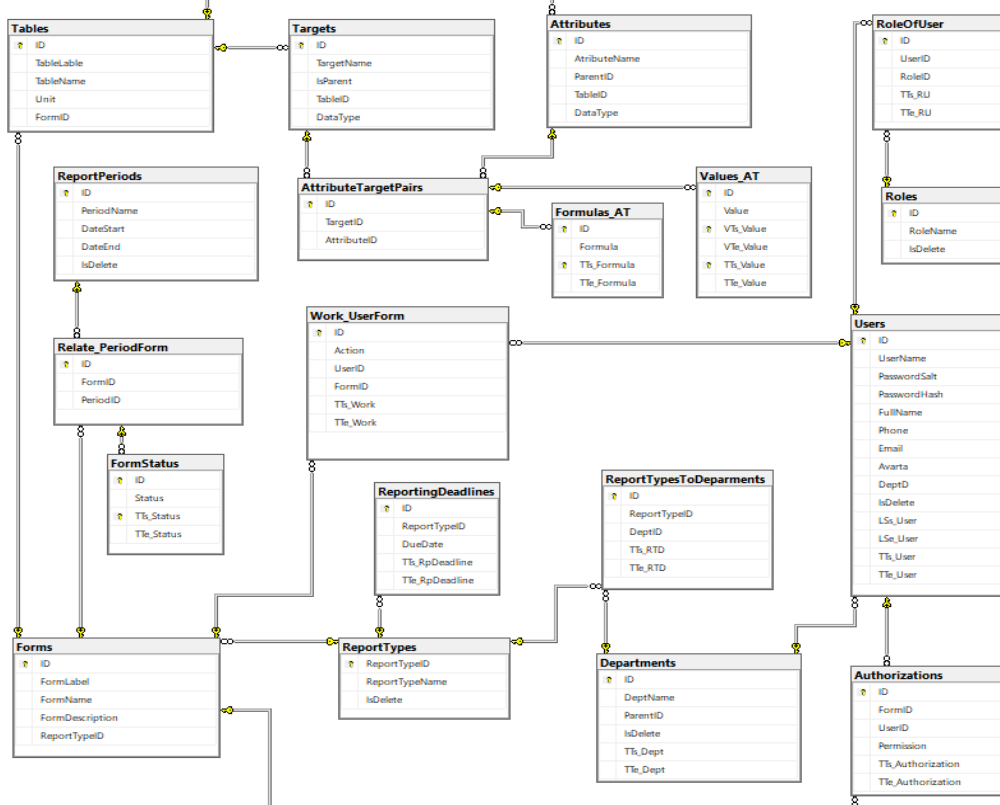
Đối với các bảng có quan hệ nhiều – nhiều, việc cài đặt CSDL thông thường sẽ phải có thêm bảng trung gian. Có hai cách để xác định khóa chính của bảng trung gian. Cách làm theo nhiều lí thuyết hướng dẫn là sử dụng khóa ngoại của hai bảng làm khóa chính của bảng trung gian. Tuy nhiên, trong thực tế, cách làm này ít được sử dụng vì khó mở rộng trong trường hợp có thêm bảng mới về sau liên quan đến quan hệ nhiều – nhiều.

Do đó, cách làm thứ hai là tạo ra khóa chính riêng giúp bảng trung gian tồn tại độc lập với các bảng liên quan đến quan hệ nhiều – nhiều. Cách làm này giúp dễ mở rộng CSDL về sau, khi cần thêm bảng mới liên quan đến quan hệ nhiều – nhiều, chỉ cần thêm một khóa ngoại mới vào bảng trung gian.

Có thể thấy, việc thiết kế CSDL mức vật lí là một quá trình quan trọng và phức tạp, đòi hỏi kiến thức chuyên sâu về cơ sở dữ liệu. Tuy nhiên, khi thực hiện đúng và hiệu quả, nó sẽ giúp cải thiện hiệu suất và độ tin cậy của hệ thống.

Ngoài ra, việc thiết kế CSDL mức vật lý cũng phải đáp ứng các yêu cầu về bảo mật dữ liệu. Để đảm bảo an toàn cho dữ liệu, các bảng trong CSDL nên được cài đặt các ràng buộc và phân quyền truy cập cho người dùng. Ngoài ra, cần sử dụng các công nghệ bảo mật như mã hóa dữ liệu, kiểm tra và giám sát các hoạt động truy cập vào CSDL.

Trong tổng thể, việc thiết kế CSDL mức vật lý là một quá trình quan trọng và phức tạp trong việc xây dựng hệ thống thông tin. Nó yêu cầu kiến thức chuyên sâu về cơ sở dữ liệu và khả năng áp dụng các quy tắc chuyển đổi lược đồ quan hệ thành mô hình vật lý. Nếu được thực hiện đúng và hiệu quả, việc thiết kế CSDL mức vật lý sẽ giúp cải thiện hiệu suất và độ tin cậy của hệ thống và đảm bảo an toàn cho dữ liệu.



Hình 5. Mô hình CSDL quan hệ của Hệ thống BCTK

Bước 5. Cài đặt, kiểm thử truy vấn dữ liệu có yếu tố thời gian

- Cài đặt CSDL thời gian:

Trong bài báo này tác giả sử dụng SQL Server phiên bản 2016 để cài đặt CSDL đã thiết kế. Đối với hệ quản trị SQL Server, các phiên bản 2016 trở về sau đều có tính năng hỗ trợ lưu trữ dữ liệu theo thời gian, bao gồm: Bảng Thời gian phiên bản hệ thống (hay còn gọi là Bảng hiện tại) lưu trữ phiên bản hiện tại của các hàng trong bảng, nó được liên kết với Bảng lịch sử có cùng lược đồ. Bảng lịch sử này tự động lưu trữ các phiên bản trước của các hàng ở Bảng hiện tại đã được cập nhật hoặc xóa.

Khi tạo Bảng thời gian phiên bản hệ thống, thì Bảng lịch sử đồng thời sẽ được tạo tự động. Nếu chúng ta không chỉ định tên Bảng lịch sử làm tham số

trong SYSTEM_VERSIONING, thì nó sẽ được hệ thống tự động đặt tên theo mẫu mặc định. Do đó, để dễ quản lý chúng ta nên chỉ định luôn tên Bảng lịch sử khi tạo Bảng thời gian phiên bản hệ thống (Kurian, 2021).

Ví dụ: Tạo bảng Values_AT và bảng ValuesHistory

```
CREATE TABLE Values_AT(
  ID INT,
  Value NVARCHAR(max),
  VTs_Value DATETIME,
  VTe_Value DATETIME,
  TTs_Value DATETIME2(0) GENERATED ALWAYS AS ROW START NOT NULL
  CONSTRAINT DF_Values_TTs DEFAULT SYSUTCDATETIME(),
  TTe_Value DATETIME2(0) GENERATED ALWAYS AS ROW END NOT NULL
  CONSTRAINT DF_Values_TTe DEFAULT CONVERT(DATETIME2, '9999-12-31
  23:59:59'),
  CONSTRAINT Values_key PRIMARY KEY(ID, VTs_Value, TTs_Value),
  CONSTRAINT fk_ValuesToAttributeTargetPairs FOREIGN KEY(ID) REFERENCES
  dbo.AttributeTargetPairs(ID),
  PERIOD FOR SYSTEM_TIME(TTs_Value TTe_Value))
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.ValuesHistory,
  DATA_CONSISTENCY_CHECK = ON));
```

Trong ví dụ trên, chúng ta đã tạo một bảng Values_AT có cột TTs_Value lưu trữ thời gian bắt đầu và cột TTe_Value lưu trữ thời gian kết thúc sự hiện diện của mỗi hàng trong bảng.

Khi tạo bảng Values_AT thì Bảng lịch sử ValuesHistory của nó cũng được tự động tạo theo và có các cột tương ứng.

- **Kiểm thử truy vấn dữ liệu có yếu tố thời gian trong Hệ quản trị CSDL SQL Server:**

Xem xét bảng **Values_AT** (ID, Value, VTs_Value, VTe_Value, TTs_Value, TTe_Value). Có thể thấy, bảng này có 2 loại thời gian là thời gian hợp lệ (VTs_Value, VTe_Value) và thời gian giao tác (TTs_Value, TTe_Value).

Ngữ nghĩa của một bộ giá trị (a, b, c, d, e, f) ∈ Values_AT được hiểu là trong khoảng thời gian [e,f] Hệ thống đã lưu thông tin: “cặp chỉ tiêu, thuộc tính có mã là a nhận giá trị là b, trong khoảng thời gian [c,d]”.

*Thao tác **Thêm:** Tôi sẽ thêm giá trị của các cặp chỉ tiêu, thuộc tính trong một khoảng thời gian cụ thể bằng cách sử dụng câu lệnh chèn đơn giản như sau:

```
INSERT INTO Values_AT(ID, Value, VTs_Value, VTe_Value )
VALUES (1, 5, '2011-01-01', '2012-12-31' ),
(4, 10, '2011-01-01', '2012-12-31' ),
(7, 20, '2011-01-01', '2012-12-31' );
```

Truy vấn các bảng Values_AT, ValuesHistory để xem kết quả đã chèn bằng câu lệnh:

```
Select * from Values_AT
GO
Select * from ValuesHistory
```

Results Messages						Values_AT
ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value	
1	5	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59	
4	10	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59	
7	20	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59	

ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value	ValuesHistory
----	-------	-----------	-----------	-----------	-----------	---------------

Sau khi thực hiện câu lệnh Chọn ở trên, sẽ thấy dữ liệu đã được chèn vào Bảng hiện tại (Values_AT). Khoảng thời gian giao tác (TTs_Value, TTe_Value) đã được hệ thống tự động lưu các giá trị thời gian vào CSDL. Trong đó, TTs_Value lưu thời gian bắt đầu xảy ra giao tác và TTe_Value là thời gian kết thúc được Hệ thống lưu một giá trị mặc định '9999-12-31 23:59:59' (đánh dấu bản ghi đang được mở). Đồng thời có thể thấy rằng, không có bản ghi nào trong Bảng lịch sử (ValuesHistory) vì chưa có hành động (cập nhật hoặc xóa) nào được thực hiện trên các hàng của Bảng hiện tại.

*Thao tác **Cập nhật**: Trước khi thực hiện cập nhật, tôi tiến hành thêm mới một bộ giá trị cho cặp chỉ tiêu thuộc tính trong một khoảng thời gian khác, để thấy rằng trong CSDL thời gian một đối tượng có thể lưu trữ được nhiều giá trị theo thời gian.

```
INSERT INTO Values_AT(ID, Value, VTs_Value, VTe_Value )
VALUES (4, 15, '2013-01-01', '2014-12-31');
```

Results Messages						
ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value	
4	10	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59	
4	15	2013-01-01 00:00:00.000	2014-12-31 00:00:00.000	2022-07-20 07:46:04	9999-12-31 23:59:59	
7	20	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59	

Chúng ta thấy được kết quả sau khi chèn bộ giá trị trên, trong bảng Values_AT có cặp chỉ tiêu thuộc tính với mã định danh ID = 4 có hai bộ giá trị tương ứng với các khoảng thời gian khác nhau.

Thực hiện cập nhật giá trị của một cặp chỉ tiêu thuộc tính trong một khoảng thời gian cụ thể bằng câu lệnh sau:

```
UPDATE Values_AT
SET Value = 11
WHERE ID = 4
and VTs_Value >= '2011-01-01'
and VTe_Value <= '2012-12-31';
```

Xem kế hoạch thực thi (Execution plan), chúng ta thấy câu lệnh trên đã thực hiện cập nhật dữ liệu ở bảng Values_AT, sau đó thực hiện chèn dữ liệu vào bảng ValuesHistory.

Truy vấn các bảng Values_AT, ValuesHistory ta được kết quả hiển thị ở các bảng đó lần lượt như sau:

Results		Messages			
ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value
4	11	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-20 08:25:13	9999-12-31 23:59:59
4	15	2013-01-01 00:00:00.000	2014-12-31 00:00:00.000	2022-07-20 07:46:04	9999-12-31 23:59:59
7	20	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59

ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value
1	5	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	2022-07-19 20:48:10
1	7	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 20:48:10	2022-07-20 07:19:51
4	10	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	2022-07-20 08:25:13

Kết quả trong bảng Values_AT sau khi thực hiện câu lệnh cập nhật cho thấy, cặp chỉ tiêu thuộc tính có mã định danh ID là 4 trong khoảng thời gian (2011-01-01, 2012-12-31) đã được cập nhật giá trị Value mới là 11 và Thời gian bắt đầu giao tác cũng được hệ thống tự động thay thế bằng thời gian xảy ra việc cập nhật.

Bên cạnh đó, sự thay đổi cũng xảy ra đối với bảng ValuesHistory, đó là bản ghi chứa thông tin tồn tại ở bảng Values_AT trước khi cập nhật, được chèn vào bảng này. Trong đó, có thể thấy giá trị thời gian kết thúc của bản ghi được chèn vào ValuesHistory trùng với giá trị thời gian bắt đầu của bản ghi sau khi được cập nhật trong bảng Values_AT.

*Thao tác **Xóa**: Thực hiện xóa một bộ giá trị của một cặp chỉ tiêu thuộc tính trong một khoảng thời gian cụ thể bằng câu lệnh sau:

```
DELETE FROM Values_AT
WHERE ID = 4
and VTs_Value >= '2011-01-01'
and VTe_Value <= '2012-12-31';
```

Truy vấn hai bảng Values_AT, ValuesHistory ta được kết quả hiển thị ở các bảng đó lần lượt như sau:

Results		Messages			
ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value
4	15	2013-01-01 00:00:00.000	2014-12-31 00:00:00.000	2022-07-20 07:46:04	9999-12-31 23:59:59
7	20	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	9999-12-31 23:59:59

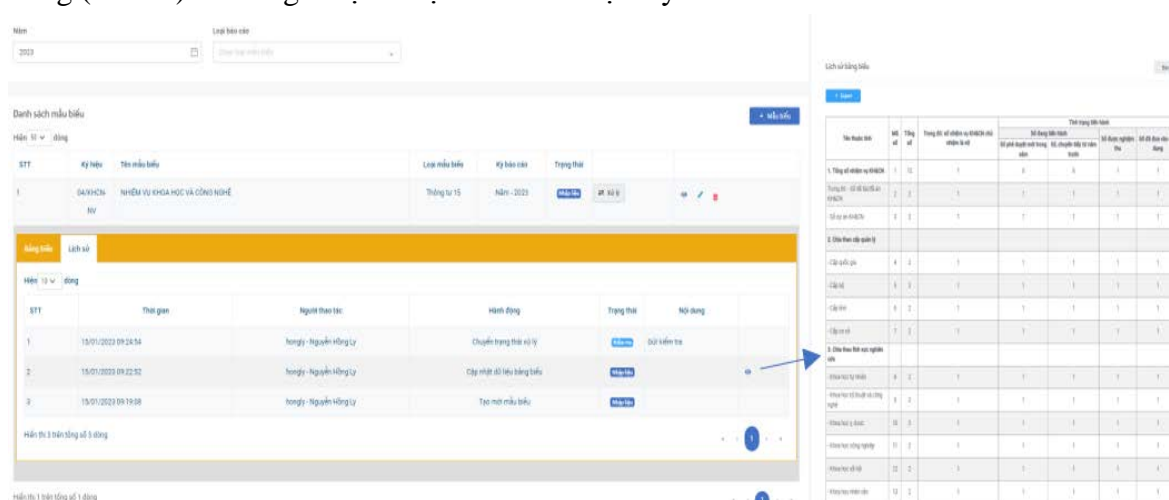
ID	Value	VTs_Value	VTe_Value	TTs_Value	TTe_Value
1	5	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	2022-07-19 20:48:10
1	7	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 20:48:10	2022-07-20 07:19:51
4	10	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-19 19:30:59	2022-07-20 08:25:13
4	11	2011-01-01 00:00:00.000	2012-12-31 00:00:00.000	2022-07-20 08:25:13	2022-07-20 09:35:26

Kết quả cho thấy, bản ghi được xóa khỏi Bảng hiện tại đã được chèn vào Bảng lịch sử và Thời gian kết thúc được cập nhật trùng với thời gian bắt đầu giao dịch, đánh dấu bản ghi được xóa đã đóng.

Như vậy, sau khi thực hiện các câu lệnh cập nhật hoặc xóa, chúng ta sẽ thấy điều đó bị ảnh hưởng trên cả hai bảng {Bảng hiện tại, Bảng lịch sử}.

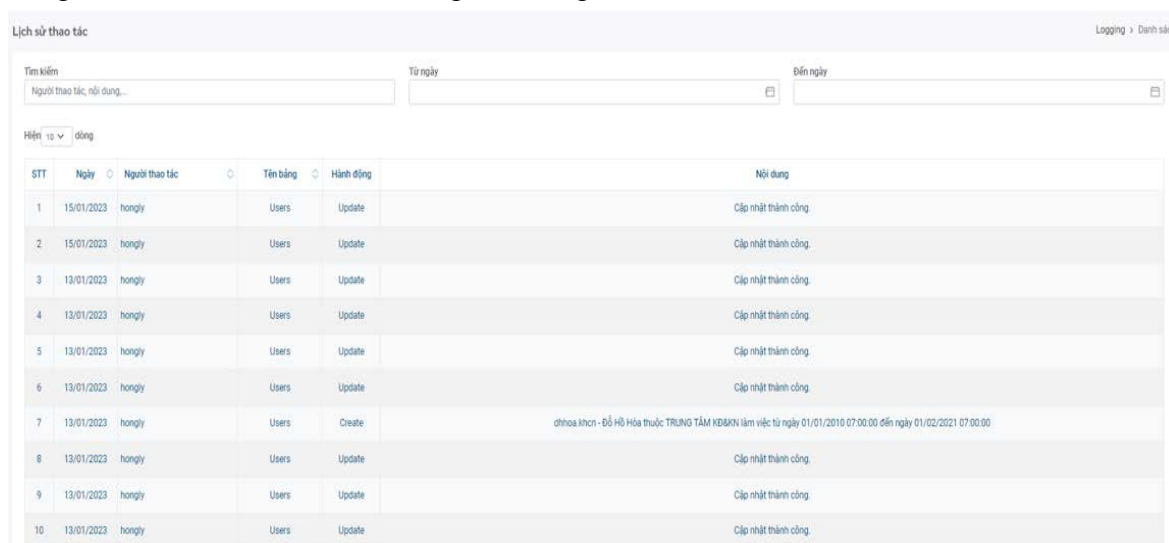
b) Một số chức năng Hệ thống BCTK sử dụng dữ liệu có yếu tố thời gian

- **Chức năng theo dõi lịch sử dữ liệu báo cáo của Hệ thống BCTK:** Trong quy trình thực hiện hoạt động báo cáo, các biểu mẫu được luân chuyển qua lại giữa các vai trò người dùng khác nhau, các giá trị (value) dữ liệu trong biểu mẫu có thể sẽ bị chỉnh sửa. Do đó, chức năng này được tác giả thiết kế để hỗ trợ các người dùng theo dõi lịch sử dữ liệu báo cáo mà bản thân họ tham gia xử lý. Cụ thể các dữ liệu được truy xuất như: thời gian giao dịch, người thao tác, hành vi thực hiện, trạng thái chuyển đổi, nội dung phản hồi của người dùng (nếu có) và các giá trị dữ liệu trước khi bị thay đổi.



Hình 6. Chức năng theo dõi lịch sử dữ liệu báo cáo

- **Chức năng theo dõi lịch sử thao tác của người dùng trên Hệ thống BCTK:** Các hành vi của người dùng đối với các dữ liệu quan trọng trên Hệ thống cần được theo dõi, giám sát. Vì vậy chức năng này được tác giả thiết kế để thực hiện ghi vết các thao tác của người dùng trên các thực thể dữ liệu trong Hệ thống.



Hình 7. Chức năng theo dõi lịch sử thao tác của người dùng trên Hệ thống

4. Kết luận

Như vậy, bài báo này đã đề xuất một giải pháp thiết kế CSDL thời gian cho một Hệ thống thông tin, từ việc phân tích bài toán cụ thể trong thực tế đến việc thiết kế và cài đặt CSDL. Đặc biệt trong đó tác giả đã chia sẻ các kinh nghiệm để vận dụng lí thuyết các phương pháp thiết kế CSDL sao cho mang lại hiệu quả. Qua đó, giúp cho người tham khảo có thể hiểu tường tận một số kiến thức về CSDL thời gian, đồng thời có thêm những kinh nghiệm để giải quyết tốt các bài toán tương tự trong thực tiễn.

Qua bài báo này, tác giả muốn chia sẻ một số kiến thức và kinh nghiệm mà bản thân tích lũy được trong thực tiễn học tập và nghiên cứu đến các bạn sinh viên ngành công nghệ thông tin (hiện nay ở cấp bậc đại học ngành công nghệ thông tin, nhiều trường ở nước ta chưa dạy về môn học CSDL thời gian).

Mặc dù, CSDL là một thành tố tính chất nền tảng quan trọng. Tuy nhiên, xây dựng CSDL chỉ là một nội dung trong nhiều công việc cần thực hiện để phát triển được hệ thống thông tin quản lí. Do trong một giới hạn bài báo khó có thể trình bày được nhiều nội dung, vì vậy sắp tới tác giả sẽ tiếp tục chia sẻ việc cài đặt thực nghiệm CSDL thời gian trên các hệ quản trị CSDL khác nhau, qua đó sẽ có những thảo luận, đánh giá và lựa chọn hệ quản trị CSDL phù hợp với từng loại dự án, đồng thời cung cấp thêm một số kiến thức, kinh nghiệm trong phân tích thiết kế hệ thống, lập trình phần mềm với CSDL thời gian đến bạn đọc.

❖ **Tuyên bố về quyền lợi:** Tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.

TÀI LIỆU THAM KHẢO

- Gregersen, H., & Jensen, C. S.(1999). Temporal EntityRelationship Models – a Survey. *IEEE*, 11(3) 464-497.
- Hoang, Q., & Nguyen, V. C.(2012). *Mot cach tiep can trong viec thiet ke Co so du lieu Thoi gian [An approach to designing Temporal Database]*. Hue University, 74A(5), 99-107.
- Kurian, L.(2021). Bi-Temporal implementation inrelational databasemanagement systems: MS SQL Server. *International Journal of Computer Science and Information Security (IJCSIS)*, 19(3).
- Jensen, C. S, & Snodgrass, R. T. (1999). Temporal Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 11(3), 464-497.

KNOWLEDGE AND EXPERIENCE IN TEMPORAL DATABASE DESIGN

Nguyen Hong Ly

Dong Thap Provincial Standards, Metrology and Quality Department, Vietnam

Corresponding author: Nguyen Hong Ly – Email: dongthap.hongly@gmail.com

Received: July 25, 2022; Revised: February 22, 2023; Accepted: February 23, 2023

ABSTRACT

Limited research has been conducted on the application of temporal databases. The review of related literature shows that articles are mainly theoretical with few applications. Therefore, it is necessary to suggest detailed solutions so that individuals and organizations can easily and effectively apply them in building a database to ensure the completeness and accuracy of information.

This article shares information on a solution to build a database for a system that supports reporting and statistical templates and forms used in administrative agencies. It includes a database building process with steps, including the analysis of some common errors related to the "mechanical" application of design methods. At the same time, the paper proposes suitable solutions to improve the efficiency of database processing to meet different usage needs in practice.

Keywords: database; design; temporal database