***Chương 1:***

**Giới thiệu**

Hệ thống quản lý cơ sở dữ liệu (DBMS) , khía cạnh hỗ trợ thời gian khác nhau của dữ liệu được cho là tạm thời. Nhưng, điều này đòi hỏi gì, nó được xác định tốt như thế nào, nó được hỗ trợ như thế nào và có thể làm gì để khắc phục mọi thiếu sót có thể xảy ra? Đó là những ví dụ về một số câu hỏi cần được giải quyết trước khi áp dụng công nghệ mới.

Dữ liệu tạm thời và DBMS hỗ trợ cho loại dữ liệu này, là một chủ đề nghiên cứu đã được nghiên cứu nhiều trong ít nhất 30 năm qua và tiếp tục thu hút các nhà nghiên cứu chú ý ngay cả ngày nay. Nhiều sách và giấy tờ đã được xuất bản trên tài liệu này. Một tổng quan xuất bản, lỗi thời nhưng toàn diện, được đưa ra trong [1]. Hầu hết các nhà nghiên cứu đã đề xuất mở rộng mô hình quan hệ. Các đề xuất sớm nhất, như HQuel, TQuel, DM / T, TempQuery và IXQuery được đề cập trong [2]. Một số tranh cãi liên quan đến những mô hình khác nhau đã xuất hiện [3]. Someproposals thậm chí đã bị chỉ trích nặng nề. Ví dụ tốt nhất về điều này có lẽ được tạo ra bởi một bài báo [4] được viết bởi Darwen và Date chỉ trích cách tiếp cận TSQL2 [5]

Dường như các trại khác nhau đã được hình thành trong cộng đồng nghiên cứu cho đến ngày nay. Hai trong số này chiếm ưu thế trong cộng đồng nghiên cứu. Tuy nhiên, cả hai phe này đều đồng ý về một điều và đó là tiêu chuẩn SQL cần một phần mở rộng. Sự khác biệt lớn nhất của ý kiến ​​dường như là về sự cần thiết phải mở rộng hoặc sửa đổi chính mô hình quan hệ để đạt được hỗ trợ tạm thời. Có thể đó là một trong những lý do chính mà các nhà cung cấp cơ sở dữ liệu lớn đã rất ngại áp dụng các khái niệm tạm thời trong các sản phẩm của họ.

Tuy nhiên, một số nhà cung cấp đã bổ sung hỗ trợ tạm thời cho các sản phẩm của họ. Đại diện nổi bật nhất trong số này là Oracle, bắt đầu với các đoạn hồi tưởng [6], DB2 [7] và TeraData [8]. Họ đã làm điều đó mặc dù SQLst Chuẩn hóa không có sẵn. Với việc chấp nhận tiêu chuẩn SQL: 2011, bao gồm hỗ trợ chung [3], chúng tôi có thể hy vọng rằng ngày càng nhiều nhà cung cấp sẽ làm điều tương tự. Một trong những nhà cung cấp đã đi theo con đường này và tham gia các nhà cung cấp khác là Microsoft Server với SQL Server 2016 của họ [ 9] sản phẩm.

Tổng quan được công bố gần đây nhất, kể từ bài viết này, về DBMS và hỗ trợ của họ cho dữ liệu tạm thời được đưa ra trong [10]. Mặc dù không toàn diện nhưng bức tranh tổng thể khá tốt.

1.1.động lực

Các nhà nghiên cứu năm 1980 đã nhận ra sự cần thiết phải giải quyết các vấn đề thời gian hoặc thay đổi theo thời gian của dữ liệu [1]. Phần cứng đã trở nên rẻ hơn, không gian đĩa lớn hơn và do đó có thể lưu được nhiều dữ liệu hơn và sau đó được lưu trữ trong thời gian dài hơn. Xu hướng này rất rõ ràng. Dữ liệu được lưu trữ, một khi có sẵn, sau đó có thể được phân tích. Một số công nghệ như khai thác dữ liệu và lưu trữ dữ liệu đã xuất hiện trong năm 1990 và dữ liệu lớn vào năm 2000.

Có nhiều trường hợp sử dụng có thể cho dữ liệu thời gian. Một số trong số họ có thể được định dạng là:

* phiên bản dữ liệu: Khi làm việc trên một bài viết, hệ thống quản lý nội dung được yêu cầu cung cấp khả năng cho người dùng quay lại các phiên bản hàng đầu của bài viết.
* Kiểm toán: Các hệ thống tạp chí y tế được yêu cầu cung cấp lịch sử về "ai" đã làm "cái gì" và "khi nào" cho mục đích kiểm toán.
* Lập kế hoạch / lập kế hoạch: Cần có phần mềm lập lịch để không cho phép sử dụng sổ kép.
* Thống kê: Công ty cần phần mềm để đo lường sự thành công của doanh số bán hàng trực tuyến của họ liên quan đến các chiến dịch họ đang chạy.

Nhiều ví dụ khác được liệt kê trong [7], với một trường hợp thú vị cho thấy rằng datahad được giữ lại trong thời gian dài hơn vì các yêu cầu pháp lý. Hơn thế nữa,

Dường như khách hàng cũng đang mong đợi ngày càng nhiều hơn từ các nhà cung cấp cơ sở dữ liệu tạm thời của họ khi kích thước dữ liệu được quản lý phát triển và ngày càng lớn hơn. Những khách hàng đó cũng có những nhu cầu tạm thời rất khác nhau, và do đó một số mô hình thú vị đã được quan sát và nghiên cứu và các giải pháp khả thi đã được đưa ra [11

Các trường hợp sử dụng được đề cập cho thấy rằng có thể có nhiều kịch bản và yêu cầu khác nhau. Khó khăn có thể phát sinh khi làm việc với dữ liệu thời gian nhưng sau đó cho hỗ trợ tạm thời là rõ ràng. Do đó, nếu một tổ chức hoặc công ty muốn áp dụng các công nghệ hỗ trợ dữ liệu tạm thời, việc đánh giá là cần thiết liên quan đến các khái niệm tạm thời nói chung và hỗ trợ cụ thể của họ trong DBMS. Một số người thậm chí còn đề xuất [7] rằng có nhiều giờ làm việc được lưu khi sử dụngDBMS với sự hỗ trợ tạm thời so với mã hóa một giải pháp tùy chỉnh trong nhà trong lớp ứng dụng.

* + 1. Vấn đề là gì?

Trong phần này chúng ta sẽ xem xét một số ví dụ, minh họa cho sự cần thiết phải giải quyết các vấn đề thời gian. Các bảng cơ sở dữ liệu được mô tả một cách lỏng lẻo, nhưng trọng tâm chủ yếu được giữ ở khía cạnh thời gian, vì không quá lo lắng về tổng thể, các ràng buộc, loại, v.v. Tương tự như vậy, các giá trị mẫu được hiển thị không nhất thiết phải biểu thị các giá trị của các loại cơ bản. Chúng chủ yếu được sử dụng để giải thích dễ dàng hơn các giá trị mẫu

* + 1. Ví dụ 1: Dấu thời gian cho các sự kiện được tạo và cập nhật

Chúng tôi bắt đầu ví dụ này bằng cách xem Hình 1.1). Có năm thuộc tính, và hầu hết trong số chúng nên tự giải thích. Khóa chính cho bảng này làPNO. Điều này cũng được chỉ định trong tiêu đề bảng. Thuộc tínhFEE trình bày tư cách thành viên cho mỗi người chơi. Để người đọc giải thích dữ liệu mẫu dễ dàng hơn, các giá trị cho cộtFEEare được hiển thị với ký hiệu tiền tệ được đặt trước. ThecolumnCREATED hiển thị khi một hàng đã được ghi lại trong cơ sở dữ liệu và thecolumnUPDATED khi một hàng được cập nhật lần cuối. Phần thời gian đã được bỏ qua từ đầu ra. Chúng ta hãy giả sử rằng các giá trị cho hai thuộc tính đó được đặt bởi hệ thống, tức là không phải bởi người dùng cuối.

Nhìn vào dữ liệu mẫu, chúng ta có thể thấy rằng nó chứa hai thuộc tính ngày. Các giá trị ngày cho thuộc tínhCREATED được tạo khi hàng được chèn và không bao giờ cập nhật lại, nhưng các giá trị choUPDATEDcan và sẽ thay đổi theo thời gian. sau khi FEE được cập nhật cho playerP1by giảm xuống còn $ 80.

Chúng tôi nhận thấy hai điều trong ví dụ này. Đầu tiên, giá trị choUPDATED đã được chuyển sang giá trị ngày hiện tại được đặt thành 1. Tháng 1 năm 2016. Đây là ngày hợp lệ hoặc có hiệu lực cho phí mới. Thứ hai, bằng cách cập nhậtFEEwe đã mất thông tin về khoản phí mà P1 đã trả trước đây. Trả lời các truy vấn như "Bao nhiêu lần phí thay đổi cho người chơiP1during năm ngoái?" Và "Điều gì làm lãng phí tổng thu nhập từ phí thành viên năm ngoái?" có thể được yêu cầu, nhưng cả hai truy vấn này đều không thể được trả lời chính xác. Bởi vì dữ liệu thay đổi theo cách này, thông tin thời tiền sử chắc chắn sẽ bị mất, và vì điều đó, chúng tôi sẽ cần phải giải quyết vấn đề này.

1.1.3 Ví dụ 2: Thời gian dập sự kiện

Đối với ví dụ này, bây giờ chúng ta hãy xem bảngPLAYER\_FEE\_SINCE, được hiển thị trong Hình1.3. Bảng này chỉ có ba thuộc tính. Đó làPNO, FEEandSINCE. DramtributionSINCEesents một năm mà lệ phí đã hợp lệ và có thể cập nhật cho người dùng. Khóa chính cho bảng này bao gồm các thuộc tínhPNOandSINCE, vì chúng tôi không thể có một người chơi trả phí nhiều lần trong cùng một năm. Chúng tôi muốn một lần nữa thay đổi phí choP1by giảm xuống còn 80 đô la. Giống như chúng ta đã làm ví dụ đầu tiên. Bây giờ chúng ta hãy xem một kịch bản có thể là gì.

Ví dụ này minh họa một giải pháp khả thi cho các vấn đề truy vấn mà chúng tôi đã xác định trong 1.1.1. Chúng tôi đã thay thế câu lệnh cập nhật bằng một hàm chèn và xác định lại khóa chính. Thay đổi này dường như cho phép chúng tôi bằng cách nào đó xây dựng phí thành viên của người chơi lịch sử. Chúng tôi có thể nói rằng phí của người chơiP1 đã là 100 đô la từ năm 2012 cho đến năm 2016 và phí đó được đặt xuống 80 đô la vào năm 2016 và sẽ có hiệu lực cho đến khi thực tế này được thay đổi. Bây giờ, hãy để chúng tôi tưởng tượng rằng người chơi này không còn là thành viên và không mất phí, nhưng chúng tôi muốn lưu giữ hồ sơ về anh ta, tức là chúng tôi không muốn xóa anh ta khỏi bàn. Làm thế nào để chúng tôi làm điều này? Nó chỉ ra rằng điều này là không thể bằng cách sử dụng mô hình hiện tại. Do đó, cần phải giới thiệu hai thuộc tính cho mô hình sẽ đại diện cho một khoảng thời gian. Một khoảng thời gian với một thời gian bắt đầu và kết thúc được chỉ định.

* + 1. Ví dụ 3: Cần cho hai chiều thời gian khác nhau

Trong các ví dụ chúng tôi đã trình bày cho đến nay, mỗi ví dụ có ngữ nghĩa khác nhau liên quan đến các thuộc tínhUPDATEDandSINCE. Cái đầu tiên, dựa trênUPDATED, chỉ thị hoặc thời gian đăng nhập của một thay đổi và không được người dùng cuối cập nhật. Thứ hai, dựa trênSINCEattribution, có thể cập nhật và đã được sử dụng để trả lại các năm mà một số phí người chơi đã hợp lệ. Đôi khi cần phải hỗ trợ cả hai chiều thời gian khác nhau về mặt ngữ nghĩa bên trong có thể lấy mẫu này.

Lệ phí cho người chơi P1 đã được ghi nhận vào ngày 1 tháng 12 năm 2011 cho năm 2012 và vẫn giữ nguyên cho đến năm 2016, khi nó được giảm xuống còn 80 đô la. Điều này được ghi lại vào ngày 1 tháng 12 năm 2015. Hơn nữa, người liên lạc của người chơi này đã được thay đổi từ JO thành MA. Tuy nhiên, vào ngày 1. người chơi đã được chỉ định cho người liên lạc trước đó của mình. Thời gian hiệu lực cho phí người chơi được ngụ ý thuộc tính SINCE, tuy nhiên thông tin liên hệ phù hợp với thời gian hiệu lực cụ thể và được ngụ ý bởi thuộc tính CẬP NHẬT. Để có thể ghi dữ liệu được hiển thị, chúng tôi phải xóa thuộc tính SINCE khỏi khóa chính. Một lần nữa, chúng ta sẽ cần một mô hình khác nếu chúng ta quản lý loại dữ liệu này

* + 1. Ví dụ 4: Cần cho các giai đoạn

Bây giờ chúng ta hãy xem xét một ví dụ sử dụng hai bảng hoàn toàn tạm thời và có liên quan.TablePLAYER\_LICENSEstores người chơi được cấp phép. Giấy phép được lấy từ một gan-gan và có giá trị trong một thời gian. Người chơi chỉ có thể tham gia các câu lạc bộ nếu họ được cấp phép. Bảng thứ haiSQUADis được sử dụng để lưu trữ các mối quan hệ câu lạc bộ người chơi, có nghĩa là người chơi đang có hợp đồng trong câu lạc bộ đó trong khoảng thời gian được chỉ định. Playermust được cấp phép trong suốt thời gian hợp đồng. Các bảng được hiển thị trong hình 1.5. Các thuộc tính được sử dụng cho các khóa chính được gạch chân trong các tiêu đề bảng. Tham chiếu khóa chính (trình phát và câu lạc bộ) không được hiển thị nhưng được giả sử là tồn tại.

Bây giờ chúng ta hãy xem xét một vài trường hợp có liên quan khi làm việc với các bảng như hai.

1 này. Có nên chèn (xông P1, 2006, 2010) vàoPLAYER\_LICENSE?

2. Nếu có thể xóa (thì P1 P1, 2008, 2012) từPLAYER\_LICENSE?

3. Có nên chèn (xông hơi11, bạn có thể chèn vào phần mềm hay không?

Câu trả lời ngắn gọn là không cho tất cả các câu hỏi này và để ngăn chặn điều đó xảy ra, chúng ta sẽ cần xác định và sử dụng các khóa chính và khóa ngoài tạm thời và các ràng buộc. Nói chung, chúng tôi muốn ngăn chặn các giá trị thời gian chồng chéo trong một bảng cụ thể. trường hợp (2) chúng tôi muốn đảm bảo rằng các khoảng thời gian trong một bảng chồng lấp khoảng thời gian cho các hàng có liên quan trong một bảng khác. Vì chúng tôi không muốn một trường hợp người chơi theo hợp đồng nhưng không được cấp phép.

* 1. Phạm vi và tuyên bố vấn đề

Sự cần thiết phải giải quyết các vấn đề khi xử lý dữ liệu thay đổi theo thời gian đã thúc đẩy các yêu cầu của khách hàng trong công ty hiện tại của tôi. Hơn nữa, công bố hỗ trợ tạm thời trong SQL Server 2016 sắp tới [9], DBMS mà khách hàng của chúng tôi, là lý do chính để chọn nền tảng này để điều tra các giải pháp tạm thời và các giải pháp khả thi. Không có cân nhắc hoặc tiêu chí nào khác đã được sử dụng.

Bây giờ chúng tôi nhắc lại một số điểm được đề cập trong phần giới thiệu cho đến nay. Họ:

• Nhiều khái niệm tạm thời được nghiên cứu và mô tả kỹ trong tài liệu.

• Một số hỗ trợ tạm thời được cung cấp bởi các nhà cung cấp cơ sở dữ liệu hàng đầu.

• Tiêu chuẩn SQL: 2011 bao gồm phần hỗ trợ tạm thời.

• Khách hàng đang yêu cầu hỗ trợ cho dữ liệu tạm thời. Và xu hướng đang tăng lên.

Xem xét tất cả những điều này, có vẻ như nó sẽ có giá trị và thường hữu ích, để thử và thực hiện hỗ trợ tạm thời đầy đủ cho một cơ sở dữ liệu phi thời gian hiện có mà yêu cầu tạm thời tồn tại. Một số câu hỏi mà chúng tôi muốn quảng cáo là: "Những hạn chế của SQL Server liên quan đến hỗ trợ tạm thời là gì?", "Có thể giải quyết những hạn chế đó không?" Và cuối cùng là "Có đáng để giải quyết không?". Điểm bắt đầu quan hệ của chúng tôi được mô tả trong Chương 3.

Hy vọng là có thể cung cấp cái nhìn sâu sắc và đưa ra câu trả lời cho những câu hỏi đó từ góc nhìn của người áp dụng. Các thách thức tạm thời đã được xử lý trong không gian ứng dụng cho phần lớn cho đến nay. Bây giờ, câu hỏi là "Có cần thiết phải ưu tiên không?".

***Chương 2:***

**Khái niệm tạm thời**

**2.1 Điểm thời gian, tỷ lệ**

Khái niệm thời gian trong một thế giới cơ sở dữ liệu tạm thời có phần khác biệt so với chúng ta

sự hiểu biết về thời gian trong thế giới thực. Trong thế giới thực, chúng ta nghĩ về một

tức thời hoặc một điểm thời gian như một thứ không có thời lượng. Dành cho

ví dụ nếu ai đó có một cuộc họp lúc 10:00 vào buổi sáng, chúng ta nghĩ theo trực giác

thời điểm đó là một không có thời lượng. Trong cơ sở dữ liệu thời gian thế giới

được xem như một tập hợp hữu hạn của các phần tử rời rạc hoặc các điểm thời gian trong một số miền thời gian.

Mỗi lần tức thì có một khoảng thời gian. Thời lượng có thể là một năm, một ngày, một giờ, một

micro giây, v.v.

Miền thời gian có thể được biểu diễn bằng các số nguyên, như trong Hình 2.1 hoặc bởi các số hữu tỷ với thang đo xác định [12]. Ví dụ: chúng ta có thể chọn một kiểu số NUMERIC (8, 6) để biểu thị các điểm thời gian, giả sử, phút có tỷ lệ hỗ trợ đại diện cho micro giây. Do đó, quy mô của một số thực, trong bối cảnh này, cũng là một khoảng thời gian tức thời và còn được gọi là độ chi tiết hoặc chronon [13] trong tài liệu cơ sở dữ liệu tạm thời.

**2.2 Khoảng**

Một khoảng trong bối cảnh tạm thời được định nghĩa là thời gian giữa hai sự kiện và được biểu diễn dưới dạng một tập hợp các niên đại liền kề [13]. Một định nghĩa khác là khoảng đó là một tập hợp các điểm thời gian rời rạc và liền kề, với một điểm bắt đầu và kết thúc được chỉ định. Loại dữ liệu

đại diện cho một khoảng, mặt khác, cũng cần phải đáp ứng các thuộc tính nhất định [12].

Các thuộc tính đó là: 1. Hai toán tử đơn nguyên BEGIN và END tồn tại, mất một khoảng

như một tham số và trả về điểm đầu tiên và điểm cuối cùng tương ứng. 2. BẮT ĐẦU (i) ≤ HẾT (i).

Và 3. Toán tử nhị phân được xác định ∈ sao cho p ∈ i, khi và chỉ khi BEGIN (i) p END (i).

Thời lượng của một khoảng có thể được định nghĩa là số lượng điểm thời gian giữa

bắt đầu và điểm cuối. Trong Hình 2.1, chúng tôi đã đánh dấu một khoảng [2: 5], bao gồm

điểm {2, 3, 4, 5} và chúng ta có thể quan sát thấy rằng giá trị của bộ này bằng với khoảng thời gian như được biểu thị trên dòng thời gian cho thang đo được chỉ định.

Các khoảng không may được gọi là các khoảng thời gian trong SQL: 2011 [3] vì khoảng từ khóa

Đã được lấy. Chúng tôi sẽ sử dụng khoảng thời gian khi chúng ta đang nói về chung

khái niệm bên ngoài bối cảnh SQL cụ thể và khoảng thời gian khác. Khoảng thời gian là

còn được gọi là khoảng cách và khoảng cách thời gian [13].

Như chúng ta đã thấy, một khoảng được biểu thị bằng thuật ngữ công chứng bằng dấu ngoặc vuông

xung quanh điểm bắt đầu và điểm cuối tương ứng. Nếu điểm bắt đầu của một khoảng bằng

điểm cuối của khoảng thời gian đóng, chúng tôi nói rằng khoảng thời gian đó là 1. Nói chung

tất cả các khoảng với thời lượng đơn vị được gọi là khoảng đơn vị [12].

Có bốn loại khoảng. Định nghĩa của họ liên quan đến một số điểm p bắt đầu từ

điểm s và điểm cuối của điểm e là:

[s: e] = {p: s ≤ p ≤ e} và s ≤ e (đã đóng)

[s: e) = {p: s ≤ p <e} và s <e (đã đóng)

(s: e] = {p: s <p ≤ e} và s <e (đóng mở)

(s: e) = {p: s <p <e} và s <e (mở)

**2.3 Khái niệm thời gian**

**2.3.1 Thời gian do người dùng xác định**

Trong tài liệu cơ sở dữ liệu tạm thời [13] có ba điểm khác biệt chính liên quan đến các khái niệm khác nhau về thời gian. Đó là thời gian giao dịch, thời gian hợp lệ và thời gian do người dùng xác định.

Thời gian do người dùng xác định hoặc thời gian sự kiện là một khái niệm đã biết và đã được sử dụng trong

cơ sở dữ liệu thế giới. Nó có thể được sử dụng để đại diện cho ngày sinh nhật của một người, ngày tuyển dụng hoặc một ngày

của một quá khứ hoặc một sự kiện sắp tới. Vì đã có hỗ trợ cho loại thời gian này, nó sẽ

không được thảo luận thêm. Do đó, trong phần tiếp theo, chúng tôi sẽ tập trung vào giao dịch và chỉ có thời gian hợp lệ.

**2.3.2 Thời gian giao dịch**

Thời gian giao dịch là thời gian khi một thực tế được lưu trữ trong cơ sở dữ liệu. Nó chỉ đề cập đến

quá khứ Các bảng hỗ trợ thời gian giao dịch cũng được gọi là bảng lịch sử, vì chúng

lưu giữ hồ sơ lịch sử của bất kỳ sửa đổi dữ liệu có thể đã được thực hiện. Chúng ta có thể

nói rằng bên cạnh việc hỗ trợ truy cập vào trạng thái hiện tại của các bảng thời gian giao dịch dữ liệu

cũng hỗ trợ truy cập vào các trạng thái hoặc phiên bản trước của dữ liệu. Nó không thể

sửa đổi thời gian giao dịch. Không có sự thay thế cho lịch sử là có thể. Thời gian giao dịch còn được gọi là thời gian hệ thống.

**2.3.3 Thời gian hợp lệ**

Thời gian hợp lệ đại diện cho thời gian khi chúng ta tin rằng một số thực tế là đúng. Nó thường được sử dụng

để đại diện cho quá khứ nhưng nó cũng có thể được sử dụng để đại diện cho thời gian hiện tại và tương lai. Như

niềm tin của chúng tôi có thể thay đổi để có thể thời gian hợp lệ. Chúng ta hãy xem xét một ví dụ. Chúng tôi ghi lại trong

cơ sở dữ liệu mà một số sự kiện lịch sử đã diễn ra trong khoảng thời gian từ năm 1956 đếnnăm 1960. Tại một số điểm sau đó chúng ta biết rằng điều này không đúng và thực tế đó là sự kiện

diễn ra từ năm 1955 đến 1960. Do đó, cần phải sửa nó và thực hiện

thay đổi trong cơ sở dữ liệu. Một ví dụ khác có thể là kế hoạch nghỉ phép của chúng tôi thay đổi và

do đó, chúng ta cần thay đổi thời gian dự kiến sang một số kế hoạch khác. Thời gian hợp lệ cũng gọi là ứng dụng hoặc thời gian kinh doanh.

**2.3.4 Bảng bitemporal**

Chúng ta đã thấy trong hai phần trước rằng giao dịch và thời gian hợp lệ đại diện cho hai

ngữ nghĩa thời gian khác nhau, trực giao với nhau. Một đại diện cho

thời gian thay đổi trạng thái bên trong cơ sở dữ liệu và thời gian khác biểu thị thời gian khi một số

thực tế được tổ chức là đúng trong thế giới mô hình.

Các bảng hỗ trợ cả hai kích thước thời gian này được gọi là bitemporal. Hình 2.2 đại diện cho sự thay đổi thời gian hợp lệ của một thực tế so với thời gian giao dịch. Hãy để chúng tôi nói rằng

thời gian hiện tại là 7, sau đó chúng ta có thể quan sát rằng tại thời điểm giao dịch 1, viết tắt là TT

hệ thống đã ghi lại rằng thời gian hợp lệ, viết tắt là VT, có khoảng [2: 3]. Hơn nữa,

tại TT = [3: 4] chúng ta có thể quan sát thấy VT = [1: 2] và tại TT = [6: 6] thời gian hợp lệ là

không được chỉ định, chỉ ra rằng thông tin thời gian hợp lệ cho một thực tế đã bị xóa tại thời điểm

6. Các sắc thái khác nhau của màu xám trong Hình 2.2 biểu thị các giá trị khác nhau có giá trị trong thời gian

khoảng thời gian hợp lệ.

Một cách khác để biểu diễn dữ liệu bitemporal dĩ nhiên là ở dạng bảng như trong thời gian khi lệ phí được thay đổi.

Xin lưu ý rằng không có ràng buộc khóa chính được xác định. Chúng tôi sẽ thảo luận về chúng sau

trong chương này. Mỗi hàng được hiển thị trong Hình 2.3 thể hiện hoặc nêu thực tế về trình phát

lệ phí cho thời gian hợp lệ. Do đó, không được có bất kỳ khoảng thời gian chồng chéo. Chồng chéo và một số vị từ tạm thời khác cũng sẽ được thảo luận sau.

**2.4 Khái niệm về bây giờ và cho đến khi thay đổi**

Bây giờ là một biến thời gian được quản lý bởi DBMS và chỉ ra rằng một thực tế là hợp lệ cho đến khi

thời gian hiện tại. Điều đó có nghĩa là, ví dụ, cho biến khoảng [3: now] bây giờ là

đánh giá và được biết đến khi người dùng yêu cầu. Nếu thời gian hiện tại là 10 thì

khoảng thời gian được ước tính đến [3:10]. Điều đó ngụ ý rằng biến bây giờ bị ràng buộc với hiện tại

thời gian hệ thống, và do đó thay đổi khi thời gian tiến triển. Trong một số cơ sở dữ liệu tạm thời

văn học [12] nó còn được gọi là "điểm di chuyển bây giờ".

Biến thời gian UC, có nghĩa là cho đến khi thay đổi là một điểm đánh dấu được sử dụng cho giao dịch

thời gian và chỉ ra hàng nào đại diện cho trạng thái hiện tại của một thực tế. Khi một trạng thái hiện tại

được thay đổi sau đó kết thúc hoặc điểm được đặt thành thời gian hiện tại và một hàng mới được đánh dấu

với UC. Nếu chúng ta nhìn vào hình 2.3, chúng ta có thể thấy rằng không có hàng hiện tại như tất cả

chúng đã bị "đóng cửa". Thời gian "đóng" này được biểu thị bằng cột SYS\_TO.

Lấy ví dụ tương tự từ Hình 2.3, bây giờ chúng ta có thể thêm một hàng bổ sung (P1, $ 10, 5,

bây giờ, 6, UC). Hàng này hiện quy định rằng phí cho người chơi P1 là 10 đô la từ lúc 5 đến nay

thời gian và nó được ghi lại tại thời điểm 6 và thực tế này là hiện tại cho đến khi thay đổi. Điều này

thay đổi được hiển thị trong Hình 2.4.

Điểm đánh dấu UC thường được biểu thị bằng một giá trị tối đa của kiểu dữ liệu lót.

Đối với kiểu dữ liệu đại diện cho năm có thể là 9999, đối với ngày, nó có thể là giá trị của

9999-12-31 và cứ thế. Đối với các ví dụ được sử dụng ở đây, nó được lấy là 99.

Tuy nhiên, các khái niệm UC và bây giờ có một số biến dạng ngữ nghĩa [14]. Ví dụ,

lấy mẫu đã đề cập trước đó, nếu thời gian hiện tại là 10 và chúng tôi phải hỏi "Mà

Người chơi sẽ bị tính phí $ 10 tại thời điểm 11? ", chúng tôi sẽ không nhận được câu trả lời chính xác. Điều đólà bởi vì biến bây giờ sẽ được ước tính thành 10. Nếu chúng ta đồng thời, để hỏi

Câu hỏi về việc người chơi P1 đã bị tính phí $ 10 bao lâu, chúng tôi sẽ lại nhận được sai

câu trả lời. Thời lượng của giai đoạn [6: 99] không thể hiện thời lượng thực tế liên quan

cho câu hỏi chúng tôi đang hỏi

**2.5 Toán tử khoảng**

Trong thế giới cơ sở dữ liệu tạm thời, có rất nhiều toán tử khoảng có thể được sử dụng

để hình thành các vị từ tạm thời. Một số toán tử sẽ được thảo luận ở đây được thực hiện

từ ngày et al. [12]. Họ chủ yếu dựa trên các nhà khai thác của Allen [15], nhưng ở đó

là một vài thay đổi tên và bổ sung. Một số trong số chúng được hiển thị trong Hình 2.5. Thêm

tổng quan hoàn chỉnh được đưa ra trong Phụ lục A.

Các khoảng i và j được coi là đóng. Trong thực tế, tất cả các khoảng trình bày trong này

luận án được đóng lại trừ khi có ghi chú khác. Ngoài ra những cái có thể được suy ra từ mẫu

dữ liệu. Với inter.s chúng tôi biểu thị điểm bắt đầu của khoảng và với inter.e là kết thúc.

Các toán tử bằng, chồng chéo, đáp ứng và sáp nhập là giao hoán. Nhà điều hành đáp ứng

là sự kết hợp của các nhà khai thác Allan, gặp gỡ và đáp ứng. Tương tự, chồng chéo toán tử là một

kết hợp chồng chéo và chồng chéo bởi. Toán tử hợp nhất là hoặc kết hợp

chồng chéo, chồng chéo bởi, gặp gỡ và đáp ứng bởi.

Vị từ để kiểm tra nếu hai khoảng gặp nhau (hoặc liền kề) bao gồm thang đo biến,

như có thể được quan sát trong hình 2.5. Trong các ví dụ của chúng tôi cho đến nay thang đo (hoặc độ chi tiết) có

được 1. Điều đáng chú ý là nếu các khoảng i và j được xác định là nửa mở, thì

vị ngữ có thể được xây dựng mà không có thang đo. Nó chỉ đơn giản là: (i.e = j.s) (i.s = j.e).

Trong hình 2.6 chúng tôi cũng chỉ ra một số tỷ lệ / độ chi tiết khác. Chúng được định nghĩa hơn

kiểu dữ liệu mẫu. Cần lưu ý rằng, đối với loại DATE, tỷ lệ hoặc độ chi tiết được hiển thị

như một ngày và không phải là số nguyên 1. Đó là do kiểu dữ liệu cơ bản. Dành cho

loại THỨ HAI nó được coi là một số nguyên. Mặt khác, đối với MILISECOND, nó là loại

NUMERIC (3,3) và đối với MICROSECOND, nó là loại NUMERIC (6,6)

**2.6 Liên kết**

Hoạt động liên kết có phần giống với loại bỏ trùng lặp. Khi thực hiện loại bỏ trùng lặp, chúng tôi xem xét nếu hai hàng tương đương. Hai hàng là

bằng nếu tất cả các giá trị thuộc tính của chúng là. Trong bảng tạm thời hai hàng tương đương giá trị

nếu các giá trị của các thuộc tính không dấu thời gian của chúng, còn được gọi là thuộc tính rõ ràng, là tương đương.

Coalescing là một hoạt động đơn phương. Nó lấy quan hệ tạm thời làm đầu vào và tạo ra

quan hệ chuẩn hóa hoặc đóng gói liên quan đến các giá trị khoảng của thuộc tính dấu thời gian của nó. Các

quan hệ kết quả là hợp nhất tương thích với quan hệ đầu vào [16].

Mối quan hệ tạm thời được kết hợp khi không có bộ dữ liệu tương đương giá trị sao cho

khoảng thời gian dấu thời gian là liền kề (đáp ứng) hoặc chồng chéo. Thực hiện các hoạt động liên kết trên

một mối quan hệ đã kết hợp lại không có hiệu lực [17]. Hình 2.7 cho thấy một ví dụ về

một hoạt động liên kết. Nếu phí cho người chơi P1 là 110 đô la cho các điểm thời gian I1 = {2, 3, 4} và

tương tự cho các điểm thời gian I2 = {5, 6, 7}, sau đó tập hợp tất cả các điểm thời gian có phí

hợp lệ là J = I1∪I2. Khoảng kết quả [2: 7] sau đó thu được bằng cách lấy giá trị nhỏ nhất

điểm khoảng thời gian bắt đầu và lớn nhất cho điểm kết thúc, tức là khoảng [MIN (J): MAX (J)].

Kết quả có thể được quan sát trong bảng b), từ Hình 2.7. Hoạt động này, như đã

được đề xuất, chỉ có thể được áp dụng cho các hàng tương đương giá trị và khoảng thời gian của chúng

gặp nhau hoặc chồng chéo. Nếu chúng ta tạo ra một tập hợp của hai tập hợp khác nhau thì kết quả

theo định nghĩa sẽ chứa các khoảng trống và các khoảng trống không được phép.

Bây giờ chúng ta hãy xem xét những gì sẽ xảy ra nếu hàng thứ hai từ ví dụ đang chạy của chúng ta

có các giá trị sau: (P1, $ 100, 3, 7). Hàng đầu tiên và thứ hai bây giờ chồng chéo. Và

chỉ kết hợp trên thuộc tính dấu thời gian, sẽ cho cùng một khoảng, cụ thể là khoảng

[2: 7]. Tuy nhiên, vì các hàng không tương đương giá trị, chúng tôi không được phép thực hiện

liên kết. Bây giờ, câu hỏi đặt ra, nếu một hàng như vậy được cho phép vào bảng để bắt đầu

với. Chúng tôi sẽ xem xét vấn đề này và một số vấn đề liên quan trong Phần 2.7

**2.7 Các ràng buộc chính**

Ngày và cộng sự. trong cuốn sách của họ [12] xác định ba vấn đề tiềm ẩn cần được giải quyết

khi làm việc với dữ liệu tạm thời. Các vấn đề được xác định là vấn đề: mâu thuẫn, dư thừa và cắt bao quy đầu. Bây giờ chúng ta hãy xem xét từng thứ bằng cách xem xét ví dụ đại diện cho những vấn đề đó.

Ví dụ trong Hình 2.8 cho thấy hai sự thật về người chơi P1. Điều đầu tiên nói rằng lệ phí

là 110 đô la cho các điểm thời gian 2, 3 và 4. Thứ hai nói rằng phí được đặt thành 100 đô la cho thời gian

điểm 3, 4, 5 và 6. Hai câu đó rõ ràng mâu thuẫn về điểm

3 và 4. Người đầu tiên nói rằng lệ phí cho những điểm đó là $ 110 và lần thứ hai là

lệ phí là 100 đô la. Nếu chúng ta định nghĩa một bảng cơ sở dữ liệu là một tập hợp các mệnh đề hoặc sự kiện thực sự, thì

hai tuyên bố đó sẽ vi phạm định nghĩa này.

Ví dụ thứ hai liên quan đến sự dư thừa, được thể hiện trong Hình 2.9. Ví dụ này minh họa rằng thực tế tương tự được nêu hai lần, cụ thể là phí cho người chơi P1 là 110 đô la cho thời gian

điểm 4. Vấn đề này có thể được giải quyết bằng cách kết hợp hai hàng đó thành một, nêu rõ

phí là $ 110 cho điểm [2: 5].

Hình 2.11 cho thấy hai bảng, đầu tiên đại diện cho người chơi và mối quan hệ câu lạc bộ người chơi thứ hai. Để cho

Bây giờ chúng tôi cho rằng trước khi một người chơi có thể tham gia một câu lạc bộ trong một khoảng thời gian, người chơi cần

là "hợp lệ" trong bảng người chơi trong cùng khoảng thời gian đó. Để thực thi loại yêu cầu này, DBMS phải hỗ trợ các khóa tạm thời nước ngoài. Ví dụ: nếu người chơi P1 có

đã tham gia một câu lạc bộ C1 trong thời gian [2: 4] không thể xóa hoặc sửa đổi

khoảng thời gian hợp lệ cho người chơi này theo cách vi phạm yêu cầu.

**2.8 Truy vấn nói chung**

Trong các hệ thống cơ sở dữ liệu thông thường, truy vấn được thực hiện trên trạng thái hiện tại của

cơ sở dữ liệu. Truy vấn tạm thời là khả năng truy vấn trạng thái cụ thể của cơ sở dữ liệu

liên quan đến một hoặc nhiều kích thước thời gian. Nếu chúng ta hãy xem ví dụ hiển thị trong

Hình 2.4, sau đó chúng ta có thể thực hiện các truy vấn trên cả thời gian giao dịch và hợp lệ. Một ví dụ

truy vấn về thời gian hợp lệ có thể được xác định là: "Nhận phí cầu thủ cho thời gian hợp lệ

khoảng thời gian tại thời điểm 3 ". Hoặc, một số khác," Nhận phí trung bình cho người chơi trong thời gian

khoảng thời gian hợp lệ [2: 4] ". Các câu hỏi về thời gian giao dịch có thể được xây dựng trong một

cách tương tự, nhưng họ sẽ luôn đề cập đến thời gian hiện tại hoặc quá khứ. Kết hợp cả hợp lệ và

khoảng thời gian giao dịch chúng ta có thể truy vấn trên hai chiều thời gian khác nhau. Hiệu quả

hỏi về "niềm tin" được mô hình hóa của chúng tôi về thời gian hợp lệ được tổ chức và ghi lại tại

một số thời gian giao dịch cụ thể. Lấy ví dụ hiện tại của chúng tôi, chúng tôi có thể xây dựng loại này

của truy vấn như: "Chúng tôi đã tin rằng phí là bao nhiêu cho người chơi P1, tại thời điểm giao dịch

3 trong khoảng thời gian [2: 4]? ".

**2.9 Tham gia và tổng hợp các chức năng**

Tham gia tạm thời là một loại liên kết giữa hai bảng, cả hai đều chứa dấu thời gian tương thích

thuộc tính.

Joins, nói chung, có thể được đại diện bởi tuyên bố sau [2]

R.A Q.B (2.2)

Chúng được gọi là theta tham gia trong đó R và Q là quan hệ và A và B là thuộc tính

với các loại dữ liệu tương thích. Toán tử theta dựa trên các biến vị ngữ sau / -

toán tử: =, 6 =, <,>, ≤ hoặc. Tham gia tạm thời liên quan đến các thuộc tính của dữ liệu thời gian

các loại, chẳng hạn như khoảng thời gian. Do đó, tham gia tạm thời có thể được định nghĩa là tham gia

dựa trên các toán tử khoảng. Các nhà khai thác như trước đây, chồng chéo và đáp ứng, đã được

thảo luận trong 2.5.

Để minh họa sự tham gia tạm thời, chúng ta sẽ sử dụng lại ví dụ từ Hình 2.4. Chúng tôi muốn chọn

tất cả các cặp người chơi đã hoặc sẽ trả tiền cho các thành viên tại một số thời điểm chồng chéo.

Liệt kê trong 2.1 giả sử sự tồn tại của trình tạo kiểu INTERVAL và toán tử

TỔNG QUAN trong DBMS.

1 CHỌN P1.PNO, P2.PNO TỪ

2 PLAYER\_FEE P1, PLAYER\_FEE P2

3 Ở ĐÂU P1.PNO <P2.PNO VÀ

4 INTERVAL (P1.VT\_FROM, P1.VT\_TO)

5 TỔNG QUAN INTERVAL (P2.VT\_FROM, P2.VT\_TO)

Liệt kê 2.1: Tham gia tạm thời

Sự tham gia tạm thời này có thể được biểu diễn dưới dạng tổng quát hơn như

ΠA1, A2, ... An

(σC ∧ R.interval INTERVALOP Q.interval (R × Q)) (2.3)

Trong đó C là một biến vị ngữ không tạm thời và INTERVALOP là toán tử khoảng. Nếu chúng ta biểu thị

vị ngữ tạm thời với T chúng ta có thể hình thành phép nối tạm thời thậm chí ngắn hơn.

R onT

C Q (2.4)

Các hàm tổng hợp, nói chung, có thể được chia thành chọn lọc (MIN, MAX) và tích lũy

Các hàm (AVG, SUM, COUNT). Trong ví dụ của chúng tôi từ Hình 2.4, liên quan đến phí người chơi, hàm tổng hợp có thể được sử dụng để tính SUM các khoản phí cho các mốc thời gian cụ thể. Hoặc là,

để tìm mức phí lớn nhất cho mỗi điểm thời gian bằng cách sử dụng chức năng MAX trên tất cả người chơi.

Tập hợp tạm thời có thể dựa trên sáu phạm vi thời gian khác nhau [11]. Những thời gian

phạm vi là: điểm trong thời gian, tức thời, ba chức năng cửa sổ khác nhau và người dùng

chức năng xác định. Tính tổng phí cho mỗi điểm thời gian sẽ là một ví dụ

của một điểm tổng hợp chức năng trong phạm vi thời gian. Kaufmann [11], trong luận án của mình, cho thấy

trường hợp sử dụng phổ biến nhất cho tập hợp thời gian trong hệ thống SAP là của

loại tức thời. Một ví dụ về loại tổng hợp này được đưa ra bởi Kaufmann và là

hiển thị trong một phiên bản được thông qua trong Hình 2.12.

Chương 3

Bàn và mối quan hệ

Trong chương này, chúng tôi sẽ giới thiệu các bảng và quan hệ giữa chúng tạo thành một

chạy ví dụ sẽ được sử dụng trong các chương tiếp theo làm cơ sở để thảo luận về nhiều

chủ đề tạm thời mà chúng tôi đã đề cập trong Chương 2.

Dữ liệu bảng sẽ được minh họa với một số giá trị mẫu. Như chúng ta đã làm trong Chương 2, trong một số

trường hợp các giá trị được trình bày sẽ không nhất thiết phải đại diện cho các kiểu dữ liệu lót.

Chúng tôi muốn làm cho các giá trị mẫu dễ đọc hơn và tập trung vào các khía cạnh thời gian.

Một tổng quan hoàn chỉnh về ví dụ đang chạy và sơ đồ cơ sở dữ liệu của nó và các câu lệnh tạo SQL có liên quan được đưa ra trong Phụ lục B.

Đối với tất cả dữ liệu mẫu, dấu UC được đặt thành 9999-12-31. Thời điểm hiện tại, hoặc bây giờ, là

giả định là 1. tháng 4 năm 2016 trừ khi có ghi chú khác.

**3.1 Vị ngữ**

Các hàng của bảng là các mệnh đề / câu lệnh đúng hoặc sự kiện về dữ liệu được mô hình hóa. Vì thế,

vị ngữ bảng có dạng tổng quát cho tất cả các mệnh đề. Khi tạo các bảng cơ sở dữ liệu thông thường, chúng ta thường không nghĩ về các vị từ một cách chính thức. Tuy nhiên,

liên quan đến dữ liệu tạm thời, chúng rất cần thiết và rất hữu ích để chúng ta có thể suy luận về

dữ liệu có thể trải dài trên các kích thước thời gian khác nhau. Lý do này cũng sẽ giúp chúng ta

rất nhiều trong việc xây dựng các ràng buộc cần thiết. Trong bảng mô tả theo họ do đó được sử dụng và chính thức nêu.

**3.2 Người chơi**

Người chơi bảng đại diện cho việc đăng ký người chơi trong một hiệp hội. Bảng là của

loại bitemporal. Vị ngữ cho bảng này là: "Người chơi PNO với tên NAME đã được đăng ký

và phí thành viên được đặt thành FEE trong khoảng thời gian VT\_FROM thành VT\_TO như được ghi lại trong thời gian

giai đoạn SYS\_FROM và SYS\_TO. ". Một người chơi có thể được đăng ký làm thành viên mà không cần

trả phí thành viên. Thành viên trả tiền được cung cấp các dịch vụ bổ sung. Các

Hình hiển thị một số giá trị mẫu cho bảng này.

TÊN PNO FEE VT\_FROM VT\_TO SYS\_FROM SYS\_TO

P1 Michael $ 110 2000-01-01 9999-12-31 2016-01-01 9999-12-31

P2 Anders $ 70 2002-04-15 9999-12-31 2016-01-01 9999-12-31

P3 Jeppe $ 90 2001-06-01 9999-12-31 2016-01-01 9999-12-31

P4 Vladimir $ 20 2004-10-01 2005-10-01 2016-01-01 9999-12-31

P5 John $ 20 2018-01-01 9999-12-31 2016-01-01 9999-12-31

*Hình 3.1: Dữ liệu mẫu cho bảng PLAYER.*

**3.3 Giấy phép người chơi**

Giấy phép người chơi được ghi lại cho thời gian hiệu lực của họ. Để có thể tham gia các đội, tức là đăng nhập

một hợp đồng, một cầu thủ phải có giấy phép hợp lệ do một số cơ quan có thẩm quyền cấp.

Có nhiều loại giấy phép khác nhau và chúng cũng được hiển thị. Giấy phép có thể được cấp

Chỉ dành cho người chơi đã đăng ký. Dự đoán cho bảng này là: "Người chơi đã đăng ký PNO có một

giấy phép loại TYPE trong khoảng thời gian VT\_FROM tới VT\_TO ".

LOẠI PNO VT\_FROM VT\_TO

P1 Junior 2009-01-01 2013-12-31

P2 Cao cấp 2008-04-15 9999-12-31

P3 Cao cấp 2009-06-01 9999-12-31

P4 Goldie 2010-01-01 2012-01-01

*Hình 3.2: Dữ liệu mẫu cho bảng PALYER\_LICENCE.*

**3.4 Câu lạc bộ**

Bảng câu lạc bộ lưu trữ dữ liệu về các câu lạc bộ đã đăng ký. Thuộc tính SINCE chỉ thời gian

khi câu lạc bộ đã được đăng ký Đây là trường hợp đặc biệt của bảng dựa trên sự kiện trong đó

chúng ta có thể suy ra khoảng thời gian hợp lệ. Các câu lạc bộ không thể được đăng ký một cách hợp lý vì nó có thể được thực hiện với người chơi. Bảng đại diện cho một hồ sơ lịch sử của đăng ký câu lạc bộ và là

chủ yếu được sử dụng để minh họa khả năng đại diện một phần hồ sơ lịch sử bằng cách sử dụng một

thuộc tính đơn.

CNO TÊN TÊN

Quảng trường quay số C1 1886-01-01

C1 Arsenal 1914-01-01

C2 Liverpool 1892-01-01

C3 Chelsea 1905-01-01

*Hình 3.3: Dữ liệu mẫu cho bảng CLUB.*

**3.5 Quản lý câu lạc bộ**

Bảng thời gian hợp lệ này được sử dụng cho các nhà quản lý câu lạc bộ như tiêu đề phần cho biết. Câu lạc bộ

người quản lý được đăng ký với thời hạn hợp đồng mà họ đã ký với

câu lạc bộ. Vị ngữ cho bảng này là: "Người quản lý có tên NAME đã ký hợp đồng với

câu lạc bộ CNO cho giai đoạn VT\_FROM thành VT\_TO ".

TÊN CNO VT\_FROM VT\_TO

C1 Wenger 1996-01-01 1999-12-31

C1 Wenger 2000-01-01 2011-12-31

C1 Wenger 2012-01-01 2018-01-01

*Hình 3.4: Dữ liệu mẫu cho bảng QUẢN LÝ.*

**3.6 Đội hình**

Chỉ những người chơi được cấp phép mới có thể tham gia các câu lạc bộ, nghĩa là ký hợp đồng với những câu lạc bộ đó. Điều này

là một bảng thời gian hợp lệ. Thời gian hợp lệ thể hiện thời hạn của hợp đồng. Khoảng thời gian của

một hợp đồng trong bảng này phải bằng hoặc dưới thời hạn của thời hạn cấp phép của người chơi.

Dự đoán cho bảng này là: "Người chơi được cấp phép PNO đã ký hợp đồng với câu lạc bộ CNO trong thời gian

giai đoạn VT\_FROM tới VT\_TO mà người chơi có giấy phép hợp lệ. ".

CNO CNO VT\_FROM VT\_TO

P1 C1 2010-06-01 2013-05-31

P2 C1 2008-10-01 2011 / 02-01

P4 C1 2010-01-01 2012-01-01

*Hình 3.5: Dữ liệu mẫu cho bảng SQUAD.*

Chương 4 Hỗ trợ toàn diện

Trong chương này, chúng tôi sẽ xem xét và kiểm tra hỗ trợ tạm thời trong Microsoft Server SQL Server 2016 [18]. Nó đã được phát hành vào ngày 1 tháng 6 năm 2016. Chúng tôi sẽ gọi nó là SQLServer kể từ thời điểm này. Máy chủ SQL đã được phát hành khi luận án này đang được hoàn thành và do đó không có đủ thời gian để sử dụng nó. Tuy nhiên, phiên bản cuối cùng không có bất kỳ thay đổi đáng kể nào liên quan đến hỗ trợ tạm thời được đưa vào phiên bản beta.

4.1 Hỗ trợ thời gian hệ thống

Trong ví dụ đang chạy, đã được mô tả trong Chương 3, hỗ trợ thời gian hệ thống được trình bày cho bảngPLAYER. Hỗ trợ thời gian hệ thống cũng được gọi là phiên bản trong Máy chủ Vista. Tên này có lẽ cũng gợi ý mục đích dự định của hỗ trợ thời gian hệ thống.

4.1.1 Lịch sử kích hoạt (phiên bản)

Để hỗ trợ thời gian hệ thống hoặc phiên bản, Microsoft đã cung cấp các phần mở rộng cho cả DDL và DML cho SQL Server [9]. Thuộc tính thời gian hệ thống được xác định vớiDATETIME2type (và chỉ với loại đó). Có hai trong số chúng, tạo thành một cặp, cho biết thời gian bắt đầu và kết thúc của giai đoạn. Phần mở rộng SQL FORI HỆ THỐNG cho phép người dùng xác định một khoảng thời gian và chỉ định các thuộc tính nào sẽ được sử dụng để tạo thành nó. Chu kỳ được định nghĩa là đóng mở. Khi xác định các thuộc tính thời gian hệ thống, người dùng cũng chỉ định các tiện ích mở rộng được xây dựng qua TUYỆT VỜI NHƯ ROW START và LUÔN LUÔN NHƯ ROW END rằng các giá trị thuộc tính sẽ được tạo ra như là giá trị bắt đầu và kết thúc của thời kỳ. Hơn nữa, người dùng cũng có khả năng xác định rằng các thuộc tính đó cần được ẩn đi. Ẩn các thuộc tính sẽ ngăn chúng tự động hiển thị các truy vấn không chọn \*. Tuy nhiên, nếu được bao gồm rõ ràng trong một danh sách chọn, chúng sẽ được hiển thị cho người dùng. Chúng có thể được ẩn bằng cách sử dụng phần mở rộng SQLHIDDEN. Các phần của câu lệnh tạo liên quan được hiển thị trong Liệt kê 4.1. Danh sách đầy đủ cho câu lệnh tạo trình phát được hiển thị trong Phụ lục B.

Phiên bản hệ thống được bật rõ ràng bằng cách sử dụng HỆ THỐNG\_VERSIONING = ON và tên của bảng được tạo bởi hệ thống trong đó hệ thống sẽ ghi lại mọi thay đổi cho đến bảng gốc. Bảng lịch sử cũng có thể được tạo bởi người dùng nhưng phải tương thích. Nếu tên của bảng lịch sử không được chỉ định, hệ thống sẽ tạo mẫu theo dõi được xác định trước. Bằng cách sử dụng hai bảng, hệ thống sẽ phân tách các trạng thái hiện tại và quá khứ của dữ liệu. Sau khi kích hoạt phiên bản thành công, người dùng có thể tự do sửa đổi lược đồ của bảng (trạng thái hiện tại) được phiên bản. Tuy nhiên, có một số hạn chế nhỏ. Người dùng không thể trao đổi loại thuộc tính thành loại khác không tương thích. Ví dụ: nó được phép thay đổiVARCHARtoCHAR, nhưng thay đổiINTtoNUMERIC thì không. Ngay cả khi thetable là trống rỗng. Mọi thay đổi lược đồ cho bảng đã được phiên bản cũng được tự động truyền vào bảng lịch sử. Điều này cho thấy SQL Server hỗ trợ sự phát triển lược đồ của các bảng được phiên bản, theo cách đó, hoàn toàn minh bạch cho người dùng. Người dùng không có hệ thống cung cấp khả năng sửa đổi trực tiếp dữ liệu trong bảng lịch sử. Mọi sửa đổi trong bảng được phiên bản được phản ánh trong bảng lịch sử bằng cách tạo các phiên bản trước. Việc sửa đổi có thể được thực hiện bằng cách: chèn, cập nhật và xóa các câu lệnh. Chúng tôi sẽ xem xét ngữ nghĩa của từng phần trong Mục 4.1.2. Trong sơ đồ ví dụ đang chạy, được tìm thấy trong Phụ lục B, bảng lịch sử được hiển thị dưới dạng bảng hỗ trợ mà không có khóa ngoại quan hệ với người bàn. Cần lưu ý rằng có thể vô hiệu hóa và kích hoạt phiên bản nhiều lần như người dùng muốn. Mỗi Thời gian người dùng kích hoạt phiên bản, người dùng có thể chọn một bảng khác cho lịch sử vì miễn là bảng này tương thích với nhau. Điều này cung cấp cho người dùng cơ hội phân vùng dữ liệu lịch sử trên nhiều bảng và có thể rất hữu ích cho các tập dữ liệu rất lớn, được tích lũy trong thời gian dài. Tuy nhiên, chỉ có một bảng lịch sử có thể được hệ thống kích hoạt và sử dụng tại bất kỳ thời điểm nào. SQL Server cũng có hỗ trợ tích hợp để xử lý các tập dữ liệu lớn một cách hiệu quả. Hình 4.1 cho thấy kiến ​​trúc trong bộ nhớ tạm thời mà SQL Server sử dụng để tối ưu hóa dữ liệu lịch sử hiện tại và gần đây. Các bảng được tối ưu hóa bộ nhớ nằm trong bộ nhớ và cung cấp truy cập nhanh. Ngoài ra, các giao dịch được thực hiện trên bảng được tối ưu hóa bộ nhớ tuân thủ hoàn toàn ACID.

Thông tin thêm về phiên bản có thể được tìm thấy trong tài liệu tham khảo [9], trong đó một số bổ sung được đề cập liên quan đến việc tạo các bảng được phiên bản. Quan trọng nhất: 1. Phải có các định nghĩa khóa chính cho bảng đã được phiên bản 2. Có chính xác một BÀI TOÁN CHO HỆ THỐNG HỆ THỐNG, và 3.PERIODcolumn luôn được coi là không có giá trị (ngay cả khi không được chỉ định) .

4.1.2 Sửa đổi dữ liệu Ghi chú chung trước khi chúng tôi đi vào các ví dụ khác nhau và xem xét việc thiết lập cho ví dụ chạy hiện tại. Đầu tiên, các giá trị cho các cột thời gian hệ thống hiện tại, không được sửa đổi theo bất kỳ cách nào bởi người dùng1. Thứ hai, bảng lịch sử là hoàn toàn chỉ đọc từ góc độ người dùng2. Thời gian hiện tại trong các ví dụ sau được giả định là 1. Tháng 4 năm 2016. XÁC NHẬN: Ngữ nghĩa của việc chèn dữ liệu vào bảng trình phát gần giống với cách chèn thông thường. Dữ liệu được chèn vào bảng mục tiêu và các cột thời gian được cập nhật tự động. Không có gì được ghi lại trong bảng lịch sử. Ví dụ về việc chèn dữ liệu được hiển thị trong Hình 4.2.

Nếu người dùng chỉ định các cột trong khoảng thời gian hệ thống, trong câu lệnh chèn của mình, thì các giá trị của các cột sẽ là hàm tích hợp DEFAULT.

UPDATING: Để cập nhật phí cho playerP1, câu lệnh cập nhật thông thường được sử dụng. Điều gì xảy ra bên dưới là như sau. Trước khi một hàng được cập nhật, nó được chèn vào bảng lịch sử với một khoảng thời gian điều chỉnh. Sau đó, hàng được cập nhật sẽ bị xóa và một hàng mới được chèn vào. Các khoảng thời gian cho hàng hiện tại và trước đó sẽ diễn ra, sau khi hoạt động được cam kết. Kết quả của hoạt động được minh họa trong Hình 4.3. Bởi vì chúng tôi đã khẳng định rằng thời gian hiện tại là 1. Tháng 4 năm 2016, các thuộc tínhSYS\_TO và SYS\_FROM trong các phần thích hợp được đặt thành giá trị này. Nói chung, tất cả các cập nhật đều được cho phép trong bảng hiện tại. Ngay cả việc sửa đổi các giá trị của khóa chính, trừ khi sửa đổi đó sẽ vi phạm khóa ngoại hoặc bất kỳ ràng buộc nào khác.

LỰA CHỌN: Xóa playerP1 sẽ sao chép hàng liên quan vào lịch sử, đóng dấu chấm và sau đó xóa hàng gốc khỏi bảng trạng thái hiện tại. Chúng ta đã thấy một ví dụ về kỹ thuật đóng thời gian trong Hình 4.3. Về cơ bản, điều đó có nghĩa là thời gian kết thúc của một khoảng thời gian đến thời gian hệ thống hiện tại. Hình 4.4 minh họa một hoạt động xóa.

Cần lưu ý rằng thao tác xóa như được minh họa, xem xét cơ sở dữ liệu mẫu của chúng tôi, sẽ không thành công. SQL Server thực sự sẽ ngăn chặn điều đó, bởi vì ràng buộc khóa ngoài từPLAYER\_LICENCEandSQUADtables.

4.1.3 Truy vấn

Để truy vấn các trạng thái hiện tại và quá khứ của dữ liệu được phiên bản, SQL Server cung cấp năm toán tử mới. Các toán tử đó là:

• AS OFdatetime: Toán tử được sử dụng để trả về trạng thái theo thời gian quy định.

• TỪ start\_datetime đến end\_datetime: Toán tử được sử dụng để trả về các hàng trong đó khoảng thời gian hệ thống trùng lặp với khoảng thời gian do người dùng chỉ định. Các khoảng thời gian được xử lý mở như đã mở.

• GIỮA start\_datetime VÀ end\_datetime. Tương tự như toán tử TỪ ĐẾN, với sự khác biệt mà các khoảng thời gian được so sánh là đóng kín.

• CONTAINED IN (start\_datetime, end\_datetime): Toán tử trả về các hàng trong đó khoảng thời gian là tập con của khoảng thời gian do người dùng chỉ định.

• ALL: Toán tử trả về tất cả các trạng thái hoặc các phiên bản, bao gồm trạng thái hiện tại. Ba toán tử đầu tiên được liệt kê tương ứng với các toán tử được xác định trong SQL: 2011 [3]. Hai toán tử này là SQL Server cụ thể. Hình 4.5 với mã mẫu hiển thị kết quả của việc chọn người chơi bằng cách sử dụng phần mở rộng SQLFORSYSTEM\_TIME với toán tửALL.

4.2 Hỗ trợ thời gian hợp lệ

SQL Server 2016 không cung cấp hỗ trợ tích hợp để làm việc với các phần dữ liệu thời gian hợp lệ. Trong phần này, chúng tôi sẽ thảo luận về những thách thức và khả năng khi thực hiện hỗ trợ thời gian hợp lệ. Việc triển khai của chúng tôi dựa trên cơ sở hạ tầng mở rộng và lập trình SQL Server. Chúng tôi cũng đã sử dụng khả năng tích hợp SQL Server với CLR3. Khả năng tích hợp này được mô tả trong [20]. Việc triển khai của chúng tôi dựa trên các khái niệm lý thuyết như được thảo luận trong Chương 2, về tiêu chuẩn SQL: 2011 (tính năng tạm thời) như đã thảo luận trong [3] và trên một ví dụ cụ thể về việc thực hiện như đã được trình bày trong [7].

4.2.1 Các ràng buộc khóa chính và khóa ngoài tạm thời

được đề cập trong Chương 2, các ràng buộc khóa chính và khóa ngoài thông thường không được đảm bảo rằng không có các câu lệnh tạm thời trùng lặp hoặc mâu thuẫn trong có thể xác định được. Việc thêm các phân phối vào khóa chính sẽ không tự ngăn chặn sự chồng chéo. Nếu chúng ta nhìn vào bảng trong Hình 4.6, chúng ta có thể thấy rằng playerP1 có hai giấy phép. Firstone ban hành trong khi anh ấy là thiếu niên và thứ hai khi anh ấy trở thành một cấp cao. Vấn đề là các giai đoạn cho hai loại giấy phép khác nhau trùng nhau. Hàng đầu tiên tuyên bố rằng người chơi đã ở giai đoạnJuniorin [2012: 2013], và lần thứ hai là anh ta làSiorioratthat cùng một lúc. Hai câu nói rõ ràng mâu thuẫn với nhau. Chúng ta cần ngăn chặn điều này xảy ra, và để làm điều này, chúng ta phải thêm các ràng buộc bổ sung cho định nghĩa theo tab. Các ràng buộc sẽ đảm bảo rằng thời hạn cấp phép cho cùng một người chơi không vượt quá.

Liệt kê trong 4.2 thêm một ràng buộc sẽ làm điều đó. Ràng buộc phụ thuộc vào hàm do người dùng xác định, lấy số người chơi, ngày bắt đầu và ngày kết thúc làm tham số và trả về 0 hoặc một tùy thuộc vào việc các hàng có liên quan có trùng nhau hay không. Nếu một chèn hoặc cập nhật sẽ vi phạm chức năng ràng buộc chồng chéo sẽ trả về 1 và 0 nếu không. Liệt kê cho hàm do người dùng định nghĩa FnPLOverlapsis có trong Phụ lục C.

ALTER TABLEPLAYER\_LICENCE

ADD CONSTRAINTChkPLOverlap

CHECK(dbo.FnPLOverlaps(PNO, VT\_FROM, VT\_TO) = 0)

Bây giờ chúng ta hãy xem xét vấn đề toàn vẹn tham chiếu tạm thời sau đây. Người chơi không thể ký hợp đồng với các câu lạc bộ nếu họ không có giấy phép hợp lệ cho thời hạn ký kết. Lấy ví dụ về hoạt động của chúng tôi, điều đó có nghĩa cụ thể là để một người chơi tồn tại trong bảng SQUAD, chính người chơi đó cũng phải tồn tại trong bảngPLAYER\_LICENCE và thời hạn cấp phép của anh ta phải có thời hạn mà người chơi đã ký hợp đồng. Hình4 minh họa yêu cầu này cho một người chơiP1. Yêu cầu toàn vẹn thời gian tham chiếu phải được thi hành khi người dùng sửa đổi dữ liệu trong bảng SQUAD cũng như khi dữ liệu được sửa đổi trong bảng PLAYER\_LICENCE. Ví dụ: không thể thay đổi thời hạn cấp phép cho người chơi khi người chơi ở dưới hợp đồng theo cách như vậy, tính toàn vẹn tạm thời bị vi phạm. Đặc điểm kỹ thuật của khóa thông thường từ bảng SQUAD đến bảng PLAYER\_LICENCE tham chiếu trình phát cụ thể mà không sử dụng thuộc tính là không thể. Đó là bởi vì khóa chính cho bảngPLAYER\_LICENCE được định nghĩa là tập hợp của hai thuộc tính, đó là kỹ thuật kiểm tra PNO và VT\_FROM.

Kỹ thuật kiểm tra ràng buộc và các hàm do người dùng xác định tương tự như hàm được sử dụng trong ví dụ rõ ràng không thể được sử dụng để thực thi toàn vẹn thời gian các trường hợp. Lý do rất đơn giản. SQL Server không thực thi ràng buộc CHECK khi xóa các trạng thái được thực thi. Do đó, người dùng có thể xóa trình phát được ký hợp đồng khỏi bảng PLAYER\_LICENCE mà không có bất kỳ kiểm tra nào được thực hiện. Vì vậy, để giải quyết vấn đề này kích hoạt được sử dụng. Đối với bảng PLAYER\_LICENCE, chúng tôi đã xác định các kích hoạt cho các câu lệnh chèn, cập nhật và xóa. Đối với các hợp đồng trong bảng SQUAD, việc xác định kích hoạt xóa là không cần thiết.

IFNOT EXISTS(

SELECT 1 FROM PLAYER\_LICENCE PLINNER JOIN inserted I

ONPL.PNO = I.PNO

WHERE dbo.FnIsContainedIn(I.VT\_FROM, I.VT\_TO, PL.VT\_FROM, PL.VT\_TO) = 1)

BEGIN

RAISERROR (’Contract-licence period constraint violation.’, 16, 1);

ROLLBACK TRANSACTION;

END

Liệt kê trong 4.3 cho thấy phần thân của trình kích hoạt được sử dụng để thực thi thời hạn hợp đồng yêu cầu được duy trì trong khoảng thời gian không liên quan. Định nghĩa kích hoạt đầy đủ có thể được tìm thấy trong Phụ lục C, Liệt kê C.3. Dưới đây liệt kê cho thấy hệ thống báo cáo lỗi nếu ràng buộc giấy phép hợp đồng bị vi phạm bằng cách cố gắng chèn một thời hạn hợp đồng không hợp lệ.

INSERT INTO SQUAD (PNO, CNO, VT\_FROM, VT\_TO)

VALUES(1, 1, ’2015-01-01’, ’2016-12-31’)

-- Error message (shorten)

Contract-licence period constraint violation.

Msg 3609,Level16, State 1, Line 125

The transaction ended in the trigger. The batch has been aborted.

Kích hoạt để ngăn chặn giấy phép bị xóa nếu thời gian của chúng được tham chiếu bởi các hợp đồng được thể hiện trong C.4. Kích hoạt này dựa trên thực tế là thời gian cấp phép không trùng lặp và thời hạn hợp đồng được duy trì trong thời hạn giấy phép. Dựa trên điều này, chúng tôi không thể kiểm tra nếu có bất kỳ hợp đồng nào được duy trì bởi thời hạn giấy phép đã bị xóa và nếu có lỗi thì sẽ được báo cáo và giao dịch được khôi phục.

4.2.2 Hợp nhất

Hoạt động liên kết, còn được gọi là đóng gói [12] và hoạt động gấp, được áp dụng cho các khoảng thời gian đó. Bằng cách làm như vậy, chúng ta sẽ có được những khoảng thời gian dài nhất có thể, tức là (khoảng thời gian tối đa), giá trị tương đương. Hình 4.8 hiển thị dữ liệu dựa trên bảngSQUAD, với một số hàng bổ sung. Những hàng mới này đại diện cho các hợp đồng mà câu lạc bộ C2 thu được. Bây giờ chúng tôi cho rằng chúng tôi muốn tìm các câu lạc bộ đã đăng ký cầu thủ trong thời gian dài nhất. Từ hình chúng ta có thể thấy rằng khoảng thời gian dài nhất thu được cho clubC1. Thời gian này là từ 2008-10-01 đến 2013-05-31, với thời gian khoảng 5 năm. Để có thể trả lời các loại truy vấn này, chúng ta sẽ cần một toán tử chung, sẽ đưa ra một khoảng thời gian quan hệ, trả về một dạng đóng gói của mối quan hệ đó.

Điều quan trọng cần lưu ý là bằng cách đóng gói tập hợp các khoảng thời gian, như chúng ta đã thấy trong ví dụ trước, trên thực tế chúng ta có được dạng chính tắc của tập hợp đó. Bằng cách làm như vậy, chúng tôi sử dụng số lượng hàng ít nhất để thể hiện thông tin cùng kỳ, vì một bộ đầy đủ các hàng sẽ làm. Hình 4.8 Thể hiện hoạt động đóng gói. Thực tế cho thấy hình thức chính tắc trên thực tế là sự bực bội duy nhất của tập hợp, ở dạng nhỏ gọn nhất, cũng có thể được hiển thị với sự trợ giúp của các phiên bản khác nhau. Ví dụ: với bộA = {[1: 3], [2: 4], [5: 6]}, vàB = {[1: 6]}. Tập hợp đơn vị in-tervals cho cả hai bộ là giống hệt nhau, cụ thể là: A ′ = B = {[1: 1], [2: 2], [3: 3], [4: 4], [5: 5] , [6: 6]}. Điều này ngụ ý rằng dạng chính tắc của setA trên thực tế là setB. Và nói chung, hai tập hợp các khoảng thời gian biểu diễn các điểm thời gian giống nhau nếu các tập hợp khoảng thời gian của chúng bằng nhau [12]. Việc thực hiện toán tử liên kếtPACK, cũng như việc thực hiện toán tử giải nén UNPACK, được trình bày và giải thích trong Phần 4.2. 5. Giải nén một tập hợp các khoảng là một hoạt động ngược lại, trong đó một tập hợp các khoảng mới được tạo ra sao cho các khoảng thời gian trong tập hợp đó là các khoảng thời gian2.2.

4.2.3 Sửa đổi các bản ghi thời gian hợp lệ

Trong phần này chúng ta sẽ xem xét ngữ nghĩa của việc thực hiện chèn, cập nhật và xóa trên các bảng thời gian hợp lệ. Đối với mỗi hoạt động này, chúng tôi sẽ trình bày khả năng phân tích. Thảo luận và giải pháp sẽ dựa trên bảngPLAYER\_LICENCE. Dữ liệu mẫu cho bảng này được hiển thị trong Hình 3.3. Đưa một bản ghi mới vào bảng tạm thời tương đương về mặt ngữ nghĩa với phép chèn triệu tập. Đó là, nếu tất cả các ràng buộc bảng, bao gồm cả các ràng buộc tạm thời, được thỏa mãn thì bản ghi mới sẽ được chèn vào bảng.

trong Hình 4.9, một hàng mới với các giá trị sau được chèn vào (P1, Senior, 2014-01-01,2014-12-31). Nói chung, chúng ta có thể nói rằng việc thực hiện chèn tạm thời, nếu thành công, sẽ phản hồi lại việc tăng số lượng của bảng. Cardinality sẽ được tăng thêm một. Điều đó về mặt ngữ nghĩa sẽ tương đương với một câu lệnh chèn thông thường. Bây giờ chúng ta hãy xem xét tuyên bố cập nhật sau: "PlayerP1 đã được cấp phép là Junior + trong khoảng thời gian từ2011-01-01 đến2014-05-31", Liệt kê 4.5 . Việc cập nhật này nếu được thực thi như hiện tại mà không có bất kỳ sửa đổi nào của hệ thống sẽ dẫn đến hai

hàng đang được cập nhật. Hàng đầu tiên trong khoảng thời gian khi người chơi được cấp phép là Junior và thứ hai khi người chơi được cấp phép là Người cao cấp. Đây rõ ràng không phải là mục đích và sẽ không được phép vì các ràng buộc ngăn chặn sự chồng chéo. Người dùng đáng tin cậy để có thể đưa ra một tuyên bố cập nhật như thế này, nó cần phải được sửa đổi bởi hệ thống. Trong SQL Server, ngữ nghĩa của một câu lệnh có thể được sửa đổi bằng cách sử dụng INSTEAD OF TRIGGER. Sử dụng trình kích hoạt, chúng tôi sẽ xóa mọi hàng hiện có cho playerP1 trong một khoảng thời gian nhất định và sau đó thực hiện thao tác chèn với các giá trị được cung cấp bên trong giao dịch đơn lẻ. Tuy nhiên, vấn đề với cách tiếp cận này là kết quả của quá trình cập nhật không nhất thiết là những gì người dùng dự định. Hệ thống cơ sở dữ liệu DB2 [7], DBMS hỗ trợ dữ liệu tạm thời dựa trên SQL: 2011, sẽ từ chối tuyên bố này và lỗi chồng chéo báo cáo. Trong thực tế, tất cả các câu lệnh cập nhật và xóa thông thường cũng được xử lý như vậy, ngay cả khi các thuộc tính tạm thời có liên quan

UPDATE PLAYER\_LICENCE SET

[TYPE] = ’Junior+’,

VT\_FROM = ’2011-01-01’,

VT\_TO = ’2014-05-31’

WHERE PNO = 1

Liệt kê 4.6 cho thấy ví dụ về câu lệnh cập nhật tương tự có thể được sử dụng để đạt được kết quả mong muốn trong DB2. Tuyên bố dựa trên SQL: 2011 tiện ích mở rộng được xây dựng như TRƯỚC HẠN <period> TỪ <start\_date> TO <end\_date> có thể được sử dụng một cách hiệu quả để cập nhật và xóa các hoạt động.

-- DB2 example

UPDATE PLAYER\_LICENCE

FOR PORTION OF BUSINESS\_TIME FROM ’2011-01-01’ TO ’2014-05-31’

SET TYPE = ’Junior+’

WHERE PNO = 1

Kết quả thực hiện câu lệnh từ Liệt kê 4.6 với tập dữ liệu hiện tại được hiển thị trong Hình4.10. Một lần nữa, câu lệnh SQL này chỉ được sử dụng để chứng minh hiệu quả của việc cập nhật và không được SQL Server hỗ trợ. Việc nâng cấp loại giấy phép cho playerP1 có trong ví dụ chạy của chúng tôi dẫn đến ba hàng. Nói chung, số lượng hàng sau một hàng cập nhật cho một phần giảm thời gian, tăng không thay đổi. Điều này sẽ chỉ phụ thuộc vào khoảng thời gian được chỉ định cập nhật.

Ngữ nghĩa cho hoạt động xóa tương tự như cập nhật. Chúng được phân biệt với các câu lệnh xóa tự nguyện bằng cách sử dụng tỷ lệ gia hạn thời gian. Cần lưu ý rằng kết quả của hoạt động xóa có thể giảm cũng như tăng số lượng hàng trong bảng. Nếu chúng tôi xem xét một hàng: (P1, Junior, 2009-01-01, 2013-12-31), sau đó xóaP1 trong khoảng thời gian từ 2010-01-01 đến2012-12-31 sẽ tạo ra hai hàng mới. Cụ thể (P1, Junior, 2009-01-01, 2009-12-31) và (P1, Junior, 2013-01-01, 2013-12-31). Liệt kê 4.7 cho thấy cách xóa phần ngữ nghĩa của thời gian có thể xóa đã đạt được SQL Server bằng cách sử dụng thủ tục lưu sẵn được tham số hóa.

4.2.4 Truy vấn

Trong phần này chúng ta sẽ xem xét ba truy vấn riêng biệt liên quan đến thời gian hợp lệ. Truy vấn A: Nhận tất cả các cặp người chơi đã ở cùng một đội tại một số điểm.SQL cho truy vấn này được hiển thị trong Liệt kê 4.8 và kết quả trong hình 4.11. Chúng tôi đang tham gia cùng một bảng trên id câu lạc bộ, và sau đó áp dụng một vị từ tạm thời để chỉ lấy các bản ghi đó trùng nhau. Preesatet1.PNO <t2.PNO được sử dụng để loại bỏ các hàng và hàng thừa trong đó cùng một người chơi khớp.

SELECT t1.PNO AS PNO1, t2.PNO AS PNO2

FROM SQUAD t1 INNER JOIN SQUAD t2 ON t1.CNO = t2.CNO

WHERE

dbo.FnIsOverlap(t1.VT\_FROM, t1.VT\_TO, t2.VT\_FROM, t2.VT\_TO) = 1

AND t1.PNO < t2.PNO

Biểu thức tìm các khoảng thời gian chồng lấp có thể được biểu thị bằng các thuật ngữ chung hơn:

[MAX (i.start, j.start): MIN (i.end, j.end)] (4.1)

Bây giờ chúng ta hãy xem truy vấn tiếp theo, Truy vấn B. Chúng tôi muốn có được tất cả các cầu thủ hiện đang ký hợp đồng, được hiển thị theo thời hạn hợp đồng của họ. Đó là, chúng tôi muốn tìm một cầu thủ có hợp đồng dài nhất. Truy vấn B: Nhận tất cả các cầu thủ đang hoạt động (các cầu thủ theo hợp đồng) được sắp xếp theo thời gian dài nhất mà họ ký hợp đồng. Câu hỏi này được hiển thị trong Liệt kê 4.10 và kết quả được hiển thị trong Hình 4.12.

SELECT\*,

DATEDIFF(day, VT\_FROM, VT\_TO) as NDAYS

ROM SQUAD

ORDER BY NDAYS DESC

Nếu chúng tôi muốn tìm độ dài hợp đồng dài nhất trong tất cả các câu lạc bộ mà người chơi đã chơi, thì chúng tôi có thể tổng hợp kết quả bằng cách tính tổng thời hạn hợp đồng cho mỗi cầu thủ.

Chúng tôi chuyển đến truy vấn cuối cùng có liên quan. Các câu hỏi liên quan đến đệ quy không phải là những câu dễ đọc nhất như chúng ta sẽ thấy. Và họ cũng rất khó hình thành nên biểu hiện trong SQL.Query C: Nhận tất cả các cặp cầu thủ ở cùng một đội trong cùng một năm và nhiều năm chơi khi họ chơi cùng nhau và đội họ đang chơi cho. truy vấn được hiển thị trong Liệt kê 4.11 và kết quả được hiển thị trong Hình 4.13.

Chúng tôi bắt đầu bằng cách tính thời lượng tính theo năm của mỗi hợp đồng và gán kết quả cho máy tính bảng1. Sau đó, chúng tôi lặp lại kết quả giảm một năm cho đến khi NYEARS đạt đến không. Kết quả được gán cho bảng t2. Tiếp theo, chúng tôi tính toán mỗi năm một người chơi đã được ký hợp đồng bằng cách thêm số năm vào năm bắt đầu và gán kết quả cho máy tính bảng3. Tại thời điểm này, tablet3 sẽ chứa các mục sau cho playerP1: {(P1, C1, 2010), (P1, C1, 2011), (P1, C1, 2012), (P1, C1, 2013)}. Các

Bước cuối cùng là tham gia bảng t3 vào câu lạc bộ id trong đó số năm bằng nhau và tạo ra kết quả cuối cùng.

4.2.5 Kiểu dữ liệu khoảng thời gian

Giá trị (hoặc thời gian) là loại dữ liệu không tồn tại trong Máy chủ SQL. Ngoài ra, tiêu chuẩn SQL: 2011 cũng không giới thiệu và chỉ định loại dữ liệu khoảng. Điều này sắp tới buộc người dùng phải suy nghĩ về hai thuộc tính được chọn là điểm bắt đầu và điểm cuối của một khoảng, nhưng thật không may, không có hỗ trợ cho bất kỳ hoạt động nào được cung cấp. Hơn nữa, nhiều thuộc tính rất quan trọng của một khoảng không được hiển thị hoặc người dùng rõ ràng. Các thuộc tính như loại khoảng (mở, đóng, v.v.), thời lượng của một khoảng (hoặc tỷ lệ), phần tử đầu tiên và cuối cùng của kiểu dữ liệu, nghĩa là loại sử dụng khoảng đó để thể hiện các điểm khoảng. Để khắc phục vấn đề này tôi đã tạo ra một Kiểu dữ liệu điểm và khoảng thời gian do người dùng định nghĩa trong C # bằng cách sử dụng các khả năng tích hợp CLR mà SQL Server cung cấp. Mã trong Liệt kê 4.12 hiển thị các câu lệnh tạo rút gọn cho các hàm kiểu chữ khoảng do người dùng định nghĩa để tạo các khoảng. Tạo câu lệnh lắp ráp nhập khẩu biên dịch .NET dll cung cấp các định nghĩa và triển khai cần thiết. Loại khoảng thời gian được triển khai dựa trên số nguyên và sẽ được sử dụng để thể hiện số năm trong trường hợp của chúng tôi. Điểm thời gian bị ràng buộc với các giá trị từ 1 đến 9999.

Có rất nhiều lợi ích mà khoảng cách như một loại cung cấp. Một số trong số đó là: 1. Người dùng không cần phải lo lắng về loại khoảng thời gian khi thực hiện xen kẽ, bởi vì nó được xử lý nội bộ bằng các phương thức loại. 2. Độ chi tiết hoặc tỷ lệ được xử lý tự động. 3. Hỗ trợ cho các điểm tối thiểu / tối đa và trước đó / tiếp theo trong một khoảng thời gian. 4. Tự động xử lý giá trị thay đổi. Và 5. Cho một điểm khoảng, thật dễ dàng để kiểm tra nếu điểm đó được chứa trong khoảng đó. Hiện tại 4.13 thể hiện một số trường hợp sử dụng. Báo cáo và điều hành được mô tả ngắn gọn trong các ý kiến.

Bây giờ, với kiểu dữ liệu khoảng thời gian, chúng tôi chuyển sự chú ý của chúng tôi để đóng gói các chức năng C.5 và giải nén C.6. Hai chức năng này không được thực hiện thông qua CLR, nhưng cả hai đều phụ thuộc rất nhiều vào loại khoảng thời gian mới. Chúng tôi sẽ không đi vào chi tiết thực hiện mà chỉ thảo luận về các trường hợp sử dụng có thể có cho họ. Hai chức năng này, như đã đề cập, được liệt kê trong fullin Phụ lục C. Hãy cho chúng tôi giả sử rằng chúng tôi muốn tìm phần tử tối đa một tập các khoảng. Ví dụ, chúng tôi có thể quan tâm đến việc tìm kiếm một câu lạc bộ có khoảng thời gian hợp đồng dài nhất giữa các cầu thủ hoạt động. Liệt kê 4.14 biểu thị điểm bắt đầu cho một giải pháp khả thi sử dụng hàm pack.

Các khoảng được chèn vào một bảng tạm thời, sau đó được đóng gói và kết quả không được phát. Các hàng được sắp xếp theo thời lượng của các khoảng thời gian. Khi cố gắng giải quyết vấn đề cho Truy vấn C4.13, SQL khá phức tạp với đệ quy đã được sử dụng. Bây giờ chúng ta hãy giả sử rằng chúng ta muốn tìm tất cả những năm mà playerP1 có giấy phép nhưng không có hợp đồng. Kết quả SQL cho truy vấn này

được hiển thị trong Liệt kê 4.15. Nếu chúng tôi muốn tìm khoảng thời gian dài nhất (khoảng thời gian tối đa) mà người chơi được cấp phép nhưng không có hợp đồng câu lạc bộ, chúng tôi sẽ chỉ đóng gói kết quả. Chúng tôi bắt đầu bằng cách tạo hai bảng tạm thời, một cho giấy phép và một cho các khoảng thời gian hợp đồng. và lấy chênh lệch đã đặt bằng cách sử dụng toán tử EXCEPT thông thường.

Chương 5 Kết quả và kết quả

5.1 Hỗ trợ và hạn chế

SQL Server 2016 chỉ cung cấp hỗ trợ cho khía cạnh thời gian hệ thống của hỗ trợ dữ liệu tạm thời. Việc kích hoạt phiên bản trên một bảng hiện có hoặc tạo một bảng mới với hỗ trợ lịch sử rất đơn giản và dễ sử dụng. Khả năng tự động ẩn các thuộc tính thời gian hệ thống có nghĩa là việc áp dụng công nghệ mới này sẽ dễ dàng hơn vì người dùng không bị buộc phải thay đổi mã hiện có và có thể áp dụng tính năng mới này khi cần thiết. Các câu lệnh DML đã được sử dụng trước khi bật lịch sử sẽ tiếp tục hoạt động mà không có bất kỳ sửa đổi nào sau khi lịch sử được bật. Các tính năng được cung cấp cho hỗ trợ thời gian hệ thống có vẻ phù hợp với mục đích đã định, cụ thể là cung cấp hỗ trợ phiên bản, nhưng điều đó có thể quá giới hạn đối với một số nhu cầu của người dùng. Nếu người dùng chỉ muốn theo dõi các thay đổi dữ liệu hoặc để lưu các thay đổi trong một khoảng thời gian thì sẽ rất dễ thực hiện. Truy vấn các bản ghi lịch sử cho các lát cắt thời gian cụ thể được hỗ trợ thông qua các tiện ích mở rộng FOR HỆ THỐNG. Điều đó có nghĩa là người dùng có thể dễ dàng nhìn thấy các trạng thái trước đó của dữ liệu và có thể khôi phục mọi thay đổi bằng cách sử dụng các câu lệnh SQL thông thường. Nếu người dùng muốn sử dụng các truy vấn tạm thời phức tạp hơn liên quan đến toán tử định kỳ để tổng hợp dữ liệu theo thời gian hệ thống thì chức năng đó cần phải được xây dựng bằng cách sử dụng các hàm, thủ tục và loại do người dùng định nghĩa hoặc kết hợp dễ dàng chấp nhận. Chủ yếu là vì, nhìn từ góc độ người dùng, ngữ nghĩa của các bảng tạm thời được tối ưu hóa bộ nhớ liên quan đến việc đồng bộ hóa các bản cập nhật vào đĩa được hệ thống xử lý tự động. Tuy nhiên, khả năng xung đột của hai thao tác ghi có tồn tại. Nếu chúng xung đột, một người sẽ "thắng" và người khác cần phải được gửi lại bởi người dùng. Các xung đột như vậy có thể ảnh hưởng đến hiệu suất và khả năng mở rộng của các tối ưu hóa bộ nhớ. Bởi vì Máy chủ SQL không cung cấp loại dữ liệu định kỳ, không có hỗ trợ trực tiếp cho các giai đoạn tìm kiếm như vậy. Máy chủ SQL không hỗ trợ nhiều loại chỉ mục khác có thể được sử dụng và được tối ưu hóa để quét chỉ mục dựa trên bộ nhớ nhanh. Hỗ trợ cho thời gian hợp lệ và do đó, dữ liệu bitemporal không được cung cấp bởi SQLServer và người dùng chỉ còn một mình để thực hiện hỗ trợ trên chính nó. Các khả năng mở rộng mà SQL Server cung cấp giúp ích về vấn đề này, nhưng không có hỗ trợ máy chủ tích hợp, rất khó, nếu không, không thể xây dựng hỗ trợ như được định nghĩa trong tiêu chuẩn SQL: 2011 và được đề cập trong tài liệu là đủ chung , để nó có thể được tái sử dụng trên các dự án. Ngay cả khi người dùng xây dựng hỗ trợ như vậy, nó sẽ bị ràng buộc bởi mô hình quan hệ đang được xem xét và sẽ không thể sử dụng lại nếu mô hình thay đổi. Nguyên mẫu được tạo ra bao gồm hỗ trợ cho khóa chính và khóa ngoài tạm thời. Nó đã được chứng minh rằng các kích hoạt và thủ tục lưu trữ có thể được sử dụng để đạt được hỗ trợ tạm thời bổ sung liên quan đến cập nhật và xóa dữ liệu tạm thời. Tuy nhiên, sự hỗ trợ này phù hợp với mô hình quan hệ đã được sử dụng. Kiểu dữ liệu khoảng đã được triển khai và thể hiện về bản chất là chung và có thể được sử dụng lại trên các mô hình khác nhau, nhưng hầu hết các hoạt động tạm thời sử dụng nó cũng sẽ cần kiến ​​thức về (các) bảng nơi sử dụng loại dữ liệu này. Do sự khái quát hóa đó bị mất. Với kiểu dữ liệu khoảng do người dùng xác định, nó đã được chứng minh rằng nó khá dễ dàng để cung cấp hỗ trợ cho:

1. Các phương thức Begin () và End (): trả về bắt đầu và kết thúc một khoảng thời gian tương ứng.

2. Các phương thức First () và Last (): trả về điểm đầu tiên và điểm cuối cùng có thể cho một khoảng tương ứng.

3. Thời lượng () / Đếm (): trả về thời lượng của một khoảng.

4. Prev () / Next (): lần lượt trả về điểm trước và điểm tiếp theo. Phương thứcPrev () sẽ thất bại trong một khoảng thời gian bắt đầu = First () và Next () sẽ thất bại trong một khoảng thời gian trong đó end = Last ().

5. Các toán tử khoảng như: Overlaps (), Meets (), Chứa ().

6. Các toán tử giống như tập hợp như: Intersect (), Union () và Minus (). Mà sẽ thất bại trong khoảng thời gian thích hợp.

7. Xử lý các loại khoảng khác nhau

8. Xử lý cho đến khi thay đổi giá trị

9. Xử lý các ràng buộc khoảng thời gian, chẳng hạn như: startend, và các giá trị được phép tối thiểu và tối đa.

Mã nguồn cho nguyên mẫu được mô tả trong Phụ lục D và được bao gồm dưới dạng gói zip.

Chương 6

Kết luận

6.1 Hỗ trợ

thời gian hệ thống hoặc hỗ trợ phiên bản, trong SQL Server 2016 đã được thiết kế để giải quyết các vấn đề thời gian cụ thể và theo quan điểm của tôi, họ đã thành công trong việc đó. Sử dụng các trường hợp như kiểm toán dữ liệu, trong đó cần theo dõi các thay đổi, ai thực hiện các thay đổi và khi nào. Hoặc ví dụ như phân tích thời gian, trong đó nhà phân tích có thể quan tâm đến việc tìm kiếm sự khác biệt về xu hướng theo thời gian được hỗ trợ rất tốt và có thể sẽ đáp ứng hầu hết nhu cầu của người dùng. Do đó, hỗ trợ tạm thời SQL Server 2016 chắc chắn đáng xem xét nếu một tổ chức có nhu cầu tương tự. Trong công ty riêng của mình, chúng tôi đã xác định được một số tình huống cụ thể trong đó phiên bản sẽ hoàn toàn phù hợp để giải quyết các vấn đề của chúng tôi. Từ kinh nghiệm thực hiện của riêng tôi để cung cấp hỗ trợ tạm thời cho thời gian hợp lệ, có vẻ như vấn đề thực sự khó giải quyết vì nhiều cách khác nhau dữ liệu thời gian có thể được sử dụng và sửa đổi. Tuy nhiên, dường như kiểu dữ liệu khoảng là một khối xây dựng cơ bản trong việc cung cấp hỗ trợ như vậy. Do đó, rất đáng tiếc khi tiêu chuẩn SQL: 2011 không đưa ra kiểu dữ liệu khoảng và cấu trúc hỗ trợ. Đối số đã được đưa ra trong [3], mặc dù nhu cầu được công nhận về định nghĩa của kiểu dữ liệu thời gian (khoảng là một từ khóa dành riêng trong SQL ), các cân nhắc về khả năng tương thích với ngăn xếp phần mềm hiện có đã được áp dụng. Tuy nhiên, việc giới thiệu và chấp nhận các loại dữ liệu mới, chẳng hạn như DATE / TIME / TIMESTAMP và các loại khác, thành tiêu chuẩn không có gì mới làm cho đối số được đề cập yếu hơn. Làm việc với các cặp thuộc tính chỉ định dấu bắt đầu và kết thúc của một khoảng gần giống như sử dụng ba thuộc tính để biểu thị năm, tháng và ngày thay vì sử dụng một thuộc tính duy nhất của loại DATE. Nó chỉ là không đủ tốt ít nhất là từ quan điểm trừu tượng hóa dữ liệu. Hơn nữa, nếu một kiểu dữ liệu khoảng không được xác định, thì nó có thể tạo các cấu trúc chỉ mục nhận biết khoảng thời gian tối ưu sẽ được sử dụng để tối ưu hóa dữ liệu tổng thể truy cập?

6.2 Các hướng tương lai

Kiểu dữ liệu khoảng thời gian được trình bày trong luận án này dựa trên các số nguyên và có thể được sử dụng để biểu diễn các khoảng thời gian theo năm. Nói chung sẽ hữu ích hơn nếu tôi đã triển khai giao diện khoảng để kế thừa và các kỹ thuật hướng đối tượng khác có thể được sử dụng để dễ dàng mở rộng và tạo các loại khoảng khác. Các loại khoảng dựa trên ví dụ về ngày, thời gian hoặc loại số. Hoặc có lẽ để tạo ra các trường hợp chuyên biệt hơn của các loại. Các ứng dụng tuyệt vời có thể cần phải có khả năng đại diện cho các khoảng tiền, pH hoặc nhiệt độ. Trong quá trình thực hiện đóng gói và giải nén của chúng tôi, chúng tôi đã giải quyết một vấn đề đặc biệt trong đó đầu vào là một tập hợp các khoảng mà không có thuộc tính nào khác. Các toán tử forthose thường hữu ích, chúng phải đóng gói và giải nén các hàng dựa trên sự tương đương giá trị của các hàng đầu vào. Một giải pháp khả thi cho vấn đề này là giới thiệu một loại dữ liệu mới sẽ đại diện cho một tập hợp các khoảng đơn vị, sau đó có thể được sử dụng để lưu trữ dữ liệu được đóng gói mỗi hàng