

시간지원 데이터 모델 및 집계함수에 관한 연구⁺

이인홍* · 문홍진* · 조동영* · 이완권* · 조현준*

A Study on Temporal Data Model and Aggregate Function

In-Hong Lee · Hong-Jin Moon · Dong-Young Cho · Wan-Kwon Lee · Hyun-Joon Cho

〈요 약〉

시간지원 데이터 모델은 시간 의미를 데이터 모델에 추가하여 시간에 따라 변화된 정보를 처리할 수 있는 데이터 모델이다. 시간지원 데이터 모델은 실세계에서 사건이 발생한 시간인 유효시간을 지원하는 데이터 모델과 데이터가 수록된 시간을 지원하는 거래시간 데이터 모델 그리고 거래시간과 유효시간을 모두 지원하는 이원시간 데이터 모델이 있다.

대부분의 시간지원 데이터 모델은 관계형 모델을 확장하여 시간지원 데이터를 처리할 수 있도록 설계된다. 시간지원 데이터 모델의 두 부류는 시간을 결합하는 단위에 따라 튜플 타임스탬프와 속성 타임스탬프의 두 가지 형식이 있다. 본 논문은 데이터 모델에서 시간 추가를 위한 기본적인 시간 개념과 시간지원 데이터 모델을 위한 고려사항을 나타낸다. 그리고 시간지원 데이터 모델을 지원시간에 따라 비교하였으며, 유효시간이 지원되는 시간지원 집계에 적합한 데이터 모델을 제안하였다.

* 이 논문은 95년도 전주대학교 학술연구비에 의하여 연구된 것임.

* 전주대학교 전자계산학과 교수

1. 서 론

시간 흐름에 따라 변경되는 자료의 이력을 처리할 수 있도록 시간에 대한 명확한 의미(semantics)를 데이터 모델에 포함시킬 경우 데이터베이스 이용자는 질의어를 통하여 간편하게 시간 개념을 갖는 자료에 접근할 수 있다.

지난 십여년 동안 시간지원 데이터를 지원하기 위한 여러 형태의 시간지원 데이터 모델이 연구되고 있다. 시간지원 데이터 모델의 대부분은 관계형 데이터 모델에 시간 정보를 지원하기 위해 시간지원 관계형 데이터 모델을 사용하고 있으며 일부는 객체지향 데이터 모델을 확장하여 시간지원 정보를 처리하고 있다[22]. 이들 데이터 모델의 대부분은 실세계에서 사건이 발생한 시간인 유효시간(valid time)을 지원하는 데이터 모델이다. 일부는 발생한 사건에 대한 정보를 데이터베이스에 수록한 거래시간(transaction time)을 지원하는 모델이며 적은 수이긴 하지만 두 개의 시간 즉, 유효시간과 거래시간을 모두 지원하는 데이터 모델도 있는데 이러한 모델을 이원시간(bitemporal) 데이터 모델이라 한다. 위에서 언급한 모든 데이터 모델을 시간지원 데이터 모델이라 부른다[8].

시간지원 데이터 모델은 시간 의미를 데이터 모델에 추가하여 시간지원 데이터를 처리할 수 있도록 기존의 데이터 모델을 확장하여 개념 모델을 설계한다. 가장 많이 사용되는 모델이 관계형 모델이 되며 일부는 ER 모델과 객체지향 모델 등에 시간 의미를 추가하여 모델을 설계한다. 시간지원 데이터 모델 설계시 고려할 사항은 시간의 표현과 시간의 기본 의미를 제시하여야 하고, 데이터에 시간을 추가하므로써 파생되는 문제인 시간 동형성(temporal homogeneous), 값의 일치성(value equivalence) 등을 고려해야 한다.

여러 종류의 시간지원 데이터 모델은 많은 부분에서 차이가 있는데 일반적인 시간지원 데이터 모델의 구분은 튜플 타임스탬프(tuple timestamping)

이면서 정규형(1NF)인 모델과 속성 타임스탬프(attribute timestamping)이면서 비정규형(non-1NF)인 모델의 두 형태로 나눌 수 있다. 두 모델의 특징은 다음과 같다. 첫째, 1NF으로 릴레이션을 구성할 수 있는 튜플 타임스탬프는 각 튜플마다 시간을 소유하므로 관계형 모델을 확장하여 사용할 경우 관리 측면이 용이하나 데이터의 중복이 많다. 둘째, non-1NF인 속성 타임스탬프의 데이터 모델은 튜플을 구성하는 속성마다 시간 간격을 소유한다. 속성 타임스탬프 모델은 데이터의 중복이 제거되나 관계형 모델을 확장하여 시간 모델을 구성할 경우 기존의 저장구조, 질의 변형(evolution)을 그대로 사용할 수 없다[9].

본 논문에서는 시간지원 데이터 모델을 위해 데이터 모델에 시간지원을 위한 시간의 기본 개념과 시간지원 데이터 모델을 설계하기 위해 기본적으로 언급되어야 할 조건들을 제시하고, 기존의 시간지원 데이터 모델을 유효시간 데이터 모델, 거래시간 데이터 모델, 이원시간 데이터 모델로 분류하고 비교한다. 그리고 기존의 이력 데이터베이스에서 집계 통계자료를 만들 수 있을 뿐만 아니라 과거 이력 데이터의 집계처리를 수행하므로써 데이터웨어하우징에서 통계자료도 제공할 수 있는 시간지원 집계를 위한 시간지원 집계함수 데이터 모델을 제안한다. 또한 시간지원 집계함수 모델에서 기존의 집계모델과 시간지원 집계모델을 비교하고 집계함수의 의미를 통해 시간지원 집계함수의 기능을 살펴본다.

본 논문의 구성은 2장에서 데이터 모델에서 지원되는 시간을 정의하고 이를 바탕으로 3장에서는 시간지원 데이터 모델을 분류하고 비교한다. 그리고 4장에서는 집계지원을 위한 시간지원 데이터 모델과 집계함수 모델을 살펴보고 결론을 맺는다.

2. 데이터 모델에서 시간지원

본 장에서는 시간지원 데이터 모델의 종류와

이들 데이터 모델을 구분하기 위한 중심적인 개념을 정의한다.

2.1 시간 영역

시간지원 데이터 모델에서는 기존의 데이터 모델에서 취급하지 않은 시간 의미를 추가시켜야 한다. 시간 의미를 표현하기 위해서는 시간에 대한 기본 시간 개체를 선택하고 시간의 순서를 나타내는 모델과 그에 적절한 타임스탬프를 선택해야 한다.

2.1.1 기본 시간 개체의 선택

- 시점(time points) : 대부분의 시스템들이 사용하는 방법으로 가장 기본적인 시간을 표현하기 위해서 시간을 하나의 점으로 사용한다. 이 경우 시간 간격은 간격의 시작점과 끝점의 쌍으로 표현할 수 있다.
- 시간 간격(time interval) : 지속시간을 주로 처리해야 하는 스케줄링 시스템에서 많이 사용하는 방법으로 가장 기본적인 시간 개체로 시간 간격을 사용한다. 이 경우 점 시간은 매우 작은 시간 간격으로 표현된다.

2.1.2 시간의 순서

- 선형 : 가장 많이 사용되는 방법으로 시간을 일직선으로 표현하며 현재, 미래와 과거에 관계없이 모든 시간들이 일직선상에 선후의 관계를 가진다. 그러므로 미래의 가능성은 하나만이 존재한다.
- 분기형 : 소프트웨어 공학 등에서 사용되는 방법으로 시간지원 시스템 내의 데이터는 한 기원이 되는 시간 점에 의해 분기될 수 있다. 점 시간들 사이의 순서는 분기를 기준으로 부분적으로 행해진다. 미래에 발생할 수 있는 사건에 대한 여러 가지 가능성에 대한 정의가 가능하다.

- 원형 : 시간의 흐름이 원형을 이룬다. 반복적인 사건이나 처리에 대한 모델링에 사용된다.

2.1.3 타임스탬프의 구조 정의

시간지원 데이터 모델에서는 시간을 기본 단위 표현을 위해 어떤 타임스탬프의 구조를 선택할 것인지를 결정해야 하며 다음과 같은 모델들이 존재한다[25].

- 유리수(dense model) : 시간을 유리수 Q 의 집합으로 표현한다.
- 실 수(continuous model) : 시간을 실수 R 의 집합으로 표현하며 연속적인 시간의 표현이 필요할 경우 사용된다.
- 정 수(discrete model) : 시간을 정수 Z 의 집합으로 표현하며 시간을 불 연속한 것으로 간주한다.

보통 실세계의 시간은 일반적으로 실수의 형태로 연속적인 값을 갖는다. 그러나, 인간이 인식할 수 있으며, 측정 가능한 가장 최소의 시간 단위인 크로논이라는 비연속적인 단위 값을 갖게 된다.

2.2 시간지원 데이터 모델을 위한 고려 사항

시간의 흐름에 따라 정보의 변화를 데이터 모델에서 표현하기 위해서는 시간 의미(temporal semantics)를 데이터 모델에 포함시켜야 한다. 시간지원 데이터 모델 작성시 다음과 같은 사항을 고려하여 모델을 설계해야 한다.

2.2.1 타임스탬프의 종류

시간 값은 의미적으로 세 가지 종류로 분류할 수 있다. 첫째, 단일 크로논인 시점과 연속적인 크로논의 집합인 간격 그리고 크로논의 임의의 집합인 시간 요소가 있다.

단일 사건(event)은 단일 크로논 값에 의해서 표현하며 시간 간격(interval)은 원자 속성의 시

간 값의 쌍에 의해 표현된다[9][20].

2.2.2 속성의 다양성

시간지원 데이터 모델에서 속성의 분류는 시간 의존적인 속성과 시간 독립적인 속성으로 분류할 수 있다. 시간 독립적인 속성은 관계 데이터 모델에서 릴레이션의 각 튜플을 구분하게 하는 키와 같은 속성이며, 시간 의존적인 속성은 시간에 따라 변화된 값으로 표현되는 속성이다.

예를 들어, 스키마 R이 이름과 직급으로 구성되었을 경우 이름은 시간이 흘러도 변화되지 않는 속성으로 시간 독립적인 속성이며, 직급은 시간이 지남에 따라 변화된 값을 가질 수 있으므로 시간 의존적인 속성이 된다.

관계형 모델을 확장한 시간지원 모델에서 각 튜플을 구분하는 키는 시간 독립적인 속성과 시간 값을 가진 기본 키이며 시간 값의 확장이 요구된다. 관계형 모델이 아닌 다른 모델, 예를 들면 객체지향형 모델에 시간 개념을 추가한 모델의 경우에는 surrogate라는 시스템 단위에서 관리하는 구분자에 의해서 각 객체를 유일하게 구분한다.

2.2.3 묵시적 또는 명시적 시간 단위

시간의 표현에는 묵시적인 시간 표현과 명시적인 시간 표현으로 나눌 수 있으며, 시간 표현 방법은 질의에서 데이터의 갱신, 명시적인 데이터의 출력 여부, 질의 언어의 표현 등과 관련된다.

시간 분류에서 거래시간은 묵시적인 시간 표현이며 유효시간은 명시적인 시간 표현이다. 거래시간이 지원되는 데이터 모델의 경우 사건의 발생을 시스템 단위에서 묵시적으로 관리하며 데이터의 갱신이 지원되지 않고 시간 속성을 질의어를 통해 명시적으로 접근할 수 없다. 그러나, 유효시간을 지원하는 데이터 모델은 명시적인 시간 표현으로 시간의 변화를 사용자가 직접 기록하며 시간 속성을 출력할 수도 있고 시간

속성 데이터를 갱신할 수도 있다.

2.2.4 시간 동형성

입의의 한 시간지원 튜플에 속한 모든 속성들에 생명 주기가 같을 경우 이를 시간 동형성(temporal homogeneous)이라 하며 또한 입의의 한 릴레이션에 속한 튜플들이 시간 동형성이 있을 경우 그 릴레이션도 시간 동형성이 있다.

이러한 시간 동형성은 데이터베이스의 각 객체(튜플, 릴레이션, 데이터베이스)에 시간 차원에 대한 동질성도 부여된다. 예를 들어 “유효시간에 대한 동형성”, “거래시간에 대한 동형성” 등으로 객체에 시간에 대한 동형성을 정의한다.

동형성의 동기는 동형성이 있는 릴레이션의 결과물의 timeslice는 null 값을 생성하지 않는다는 사실을 이끌어 낸다. 그러므로, 동형성 관계형 모델은 null 값을 소유하지 않은 스냅샷 관계형 모델의 부분이다. 시간 동형성은 속성단위 타임스탬프 모델에서보다 튜플단위 타임스탬프 모델에서 필요하다.

2.2.5 값의 일치성과 결합

두 개의 튜플이 특정의 시간 간격 속성을 제거한 나머지 값이 동일할 경우 두 개의 튜플은 값의 일치성(value equivalence)이 있다고 하며, <그림 1>의 (a)와 같다. 또한 시간 값이 연속적이거나 시간 값이 겹치는 현상이 발생하였을 경우 이를 값의 결합(value coalescing)이라 한다. <그림 1>의 (b)는 두 개의 튜플이 값의 일치성이 있고 값의 결합도 된다.

3. 시간지원 데이터 모델

시간지원 데이터 모델은 지원되는 시간에 따라 모델을 분류할 수 있다. 기존의 릴레이션 모델과 같이 시간 개념이 없는 모델을 스냅 데이터 모델이라 하고 실세계에서 사건이 발생한 시

간을 나타내는 유효시간 데이터 모델과 발생된 사건에 대한 정보를 데이터베이스에 수록한시간을 지원하는 거래시간 데이터 모델이 있다.

| 이름 | 직급 | 유효시간 |
|-----|-----------|----------|
| Tom | professor | [10, 13) |
| Tom | professor | [16, 20) |

(a) 값의 동등성

| 이름 | 직급 | 유효시간 |
|-----|-----------|---------|
| Tom | professor | [5, 9) |
| Tom | professor | [8, 13) |

(b) 값의 결합

<그림 1> 값의 일치성과 결합

또한 유효시간과 거래시간을 모두 지원하는 이원시간 데이터 모델이 있다[20][24]. 본 장에서는 관계형 데이터 모델을 기반으로 여러 시간지원 데이터 모델을 분류하고 각 모델의 특징을 비교한다.

3.1 개 요

시간지원 데이터 모델은 시간과 연관된 자료를 처리하기 위해 기존의 데이터 모델에 시간 속성을 추가한 데이터 모델을 사용한다. 대부분의 모델은 관계형 데이터 모델을 확장하여 사용하나 일부 객체 지향 데이터 모델과 EER (Enhanced Entity Relationship) 모델 등에 시간 속성을 추가하여 확장된 모델을 사용하기도 한다[22].

시간지원 데이터 모델의 분류는 지원되는 시간 여부에 따라 유효시간 데이터 모델, 거래시간 데이터 모델, 이원시간 데이터 모델로 나눌 수

있다. <표 1>은 대표적인 시간지원 데이터 모델로 데이터 모델의 이름과 지원되는 시간, 제안자를 나타낸 것이다(단 '-'은 데이터 모델에 대한 이름이 정의되지 않는 경우이다).

3.2 유효시간 데이터 모델

<그림 2>는 유효시간 데이터 모델에 대한 릴레이션의 예로 (a)는 Sarda 모델[15]에 의한 릴레이션으로 튜플 타임스탬프 형식이고, 시간 표현은 시간 간격을 사용하였으며 (b)는 Clifford 데이터 모델[4]로서 속성 타임스탬프 형식을 사용하며 튜플의 생명주기를 시간 요소로 표현하였고 (c)는 Tansel의 모델[21]로 속성 타임스탬프와 시간 간격으로 표현하였다.

3.3 거래시간 데이터 모델

거래시간을 지원하는 데이터 모델은 비삭제 정책을 사용한다는 것이 유효시간 데이터 모델과의 차이가 되며 데이터가 데이터베이스에 수록된 시간을 나타낸다. <그림 3>은 Jensen의 데이터 모델[8]이며, 거래시간을 나타내는 시간속성과 연산을 나타내는 속성이 있으며, 단일 크론과 튜플 타임스탬프로 표현하고 있다.

3.4 이원시간 데이터 모델

유효시간과 거래시간이 모두 지원되는 데이터 모델이다. <그림 4>는 Snodgrass의 이원시간 데이터 모델[18]으로써 유효 시작시간과 종료시간, 거래 시작시간과 종료시간을 나타내는 속성이 있으며, 시간 간격과 튜플 타임스탬프로 표현하고 있다.

3.5 시간지원 데이터 모델 비교

<표 1>에서 나타난 시간지원 데이터 모델은 아래의 4가지의 조건에 의해 각 모델을 비교할 수 있다.

- 유효시간을 표현하는 방법은?
- 거래시간을 표현하는 방법은?
- 속성에 대한 동형성(homogeneous)의 표현 여부?
- 속성에 대한 결합(coalescing)을 표현할 것인지?

<표 1> 시간지원 데이터 모델

| 데이터 모델 | 지원 시간 | 제안자 |
|----------------------------------|------------|-----------|
| Temporally Oriented Data Model | 유효시간 | Ariav |
| Time Relational Model | 거래시간, 유효시간 | Ben-Zvi |
| Historical Relational Data Model | 유효시간 | Clifford |
| TempSQL | 거래시간, 유효시간 | Gadia |
| DM/T | 거래시간 | Jensen |
| LEGOL 2.0 | 유효시간 | Jones |
| - | 거래시간, 유효시간 | Mckenzie |
| Temporal Relational Model | 유효시간 | Navathe |
| HSQL | 유효시간 | Sarda |
| Temporal Data Model | 유효시간 | Segev |
| TQuel | 거래시간, 유효시간 | Snodgrass |
| HQuel | 유효시간 | Tansel |

| Name | Course | Period |
|--------|---------|--------|
| Bill | English | 1..2 |
| Bill | English | 3..5 |
| George | English | 1..3 |
| George | Math | 5..7 |

(a) Sarda의 데이터 모델

| 튜플 값 | | 튜플 생명주기 |
|--|--|-----------|
| Name | Course | |
| 1→Bill 3→Bill 4→Bill | 1→English 3→English 4→English | {1,3,4} |
| 1→George 2→George 5→George 6→George | 1→English 2→English 5→Math 6→Math | {1,2,5,6} |

(b) Clifford 데이터 모델

| Name | Course |
|--------|---------------------------------------|
| Bill | {(1,2), English), (3,5), English)} |
| George | {(1,3), English), (5,7), Math)} |

(c) Tansel의 데이터 모델

<그림 2> 유효시간 데이터 모델

| Name | Course | Time | Op |
|--------|---------|------|-----|
| Bill | English | 423 | Ins |
| Bill | English | 427 | Mod |
| George | English | 438 | Ins |

<그림 3> 거래시간 데이터 모델

| Name | Course | Valid Time | | Transaction Time | |
|--------|---------|------------|-----|------------------|------|
| | | Begin | End | Start | Stop |
| Bill | English | 1 | ∞ | 423 | 427 |
| Bill | English | 1 | 1 | 427 | ∞ |
| George | English | 1 | 2 | 438 | ∞ |

<그림 4> 이원시간 데이터 모델

3.5.1 유효시간

시간지원 데이터 모델에서 유효시간을 표현할 경우 두 가지 사항을 고려해야 한다.

첫째, 시간 표현 방법을 결정해야 한다. 시간 표현 방법에는 시점 타임스탬프의 크로논으로 표현하는 방법, 간격 타임스탬프로 표현하는 방법과 튜플들의 집합에 시간을 결합하는 유효시간 요소(valid-time element)로 표현하는 방법이 있다.

둘째는 유효시간의 결합을 속성 타임스탬프로 표현할 것인지 튜플 단위의 시간 단위로 표현할 것인지를 결정해야 한다. <표 2>는 유효시간 데이터 모델을 위의 두 가지 조건에 의해 분류한 것이다. <표 2>에서 보는 바와 같이 시간시점에 대한 속성 타임스탬프 모델과 유효시간에 대한 튜플 타임스탬프 모델은 고려하지 않고 있다.

<표 2> 유효시간의 표현

| | 시간시점(event) | 시간 간격(interval) | 유효시간요소(time element) |
|----------|----------------|---|----------------------|
| 속성 타임스탬프 | | Mckenzie Tansel | Clifford Gadia |
| 튜플 타임스탬프 | Ariav Segev | Ben-Zvi Jones Navathe Sarda Snodgrass | |

3.5.2 거래시간

거래시간을 표현하는 방법도 유효시간을 표현하는 방법과 조건은 비슷하다. 거래시간을 표현하는 방법은 다음과 같다.

- 단일 시간단위(chronon)
- 간격: 새로 삽입되는 튜플은 시작시간은 “now”가 되며 종료시간은 until-changed가 된다.
- 세개의 크로논: Ben-Zvi가 제안한 모델로 유효 시작시간에 대한 거래시간, 유효 종료시간에 대한 거래시간과 튜플이 논리적으로 제거되는 시간으로 거래시간을 표현한다.

아래 <표 3>은 거래시간 모델을 분류한 것이다. 타임스탬프와 크로논에 의하여 비교하였으며, Gadia만 속성 타임스탬프를 이용하였다.

<표 3> 거래시간의 표현

| | 단일 chronon | 간격 (chronon의 쌍) | 세 개의 chronon | 거래시간 요소 (chronon의 집합) |
|-----------------------|---------------|--------------------|-----------------|-----------------------------|
| 속성 타임스탬프 | | | | Gadia |
| 튜플 타임스탬프 | Jensen | Snodgrass | Ben-Zvi | |
| 튜플집합 에 대한 타임스탬프 | | Mckenzie | | |

3.5.3 시간 동형성과 값의 결합 및 속성의 구조

<표 4>는 각 모델의 유효시간 동형성의 지원 여부와 각 모델에서 속성을 표현하는 방법에 관한 표로서 이들 모델들은 모두 거래시간에 대한 동형성을 지원한다. 시간지원 데이터모델 중 튜플 타임스탬프 모델들은 시간 동형성이 요구되며 시간 동형성이 지원되는 모델들의 타임스탬프는 단일 크로논을 이용한다.

또한, 시간지원 데이터 모델을 설계할 경우 중요한 결정중의 하나는 속성 값을 어떻게 표현할 것인가 하는 문제이다. 아래는 속성 값을 표현하는 방법을 나타낸 것이며 제시된 시간지원 모델들에 대한 속성 표현 방법은 <표 4>의 4번째 열에 나타난다.

- 원자 값(atomic valued) : 속성 값들이 내부적인 구조를 갖지 않는다
- 집합 값(set valued) : 속성 값들은 원자 값의 집합이다.
- 함수적 원자 값(functional, atomic valued) : 속성 값들은 시간 도메인(유효시간) 부터 속성 도메인을 나타내는 함수에 의해 표현한다.
- 순서쌍(ordered pairs) : 속성 값 들은 값과 타임스탬프의 순서쌍이다.
- Triplet valued : 속성 값의 구성은 속성 값, 유효 시작시간, 유효 종료시간의 세 개의 요소로 구성된다.

<표 4> 시간지원 데이터 모델의 비교

| 데이터 모델 | 유효시간 동질성 | 유효시간 결합 | 속성 값 |
|-----------|-------------|------------|--|
| Ariav | yes | no | atomic |
| Ben-Zvi | yes | no | atomic |
| Clifford | no | no | functional |
| Gadia | yes | no | functional |
| Jensen | - | - | atomic |
| Jones | yes | no | atomic |
| Mckenzie | no | yes | ordered pairs |
| Navathe | yes | yes | atomic |
| Sarda | yes | no | atomic |
| Segev | yes | no | atomic |
| Snodgrass | yes | yes | atomic |
| Tansel | no | no | atomic, set-valued, triplet, set-triplet |

시간변위 환경에서 데이터 모델을 고려하기 위하여 지금까지 연구된 시간개념을 검토하였고

시간개념이 포함된 시간지원 데이터 모델을 비교 정리하였다. 이 시간지원 데이터 모델에서 사용자에게 중요한 통계 제공기능인 시간지원 집계함수를 기술한다. 시간지원 집계함수는 기존의 집계함수와 동일한 함수 이름을 사용할 지라도 내부의 수행기능이 다르다. 그 이유는 시간의 변화에 따라 동일한 내용을 갖는 튜플들이 반복 생성될 수 있으므로 시간에 따른 집계기능을 수행하기 때문이다.

4. 시간지원 집계함수 데이터 모델

집계함수란 릴레이션의 한 속성에 대하여 객체들이 갖는 모든 값들에 대한 합계, 평균, 개수 등을 구하는 함수로 데이터베이스 시스템에서 제공해 준다. 그러나 시간지원 데이터 모델에서 집계 처리는 객체들이 소유한 값에 대한 집계 기능뿐만 아니라 한 객체가 시간에 따라 변화된 집계 기능도 필요하다.

본 연구의 시간지원 집계함수 데이터 모델에서는 유효시간을 지원하는 이력 데이터베이스로 한하며 유효시간의 표현은 시점과 시간 간격으로 표현하고 튜플 타임스탬프를 사용한다.

4.1 기존 집계 함수 모델

관계형 데이터 모델에서 주로 사용하는 SQL에서 지원하는 집계함수는 count, sum, min, max, avg 등이 있으며, Quel에서는 count, sum, min, max, avg, any 등 기본 집계함수와 유일한 값을 계산하기 위한 countU, sumU, avgU 등 유일(unique) 집계함수가 있다. 이러한 집계함수들은 min, max 등과 같이 특정 값을 찾아내는 선정 집계함수(select aggregate)와 sum, avg 등과 같이 계산에 의하여 결과를 산출하는 계산 집계함수(compute aggregate) 및 count, any 등과 같이 non-dimensional quantity 를 계산하는 집계함수로

분류하며 이들 집계 함수들은 다음과 같이 두 가지 형태로 분류된다[23].

- 스칼라형 집계함수(scalar aggregate) : 결과 값으로 하나의 값을 생성한다.
- 릴레이션형 집계함수(aggregate function) : 대상 릴레이션의 한 부분 집합에 대해 주어진 집계함수를 계산하여 결정되는 여러 개의 값들을 생성한다.

4.2 시간지원 집계함수 모델

시간지원 데이터 모델에서 지원하는 집계함수는 기존의 관계형 모델에서 지원하는 집계함수들과 시간 속성들을 대상으로 집계 기능을 수행하는 부가적인 집계함수들이 있다. 시간지원 데이터 모델에서 집계함수는 물리적인 측면과 논리적인 측면에 의해 <그림 5>와 같이 나눈다.

물리적인 측면의 분류에서 한 집계함수가 다른 집계함수의 내부에 존재하는 경우를 중첩된 집계함수(nested aggregate)라 하고 집계함수내에 또다른 집계함수를 포함하지 않는 경우를 단일 집계함수(single aggregate)라 한다. 대상 속성에 따라서는 기존의 데이터 모델에서 사용하는 오직 일반적인 속성 값에 대해 연산을 수행하는 집계함수를 시간 독립적인 집계함수(time-independent aggregate)라 정의하고 시간 속성을 대상으로 집계 기능을 수행하는 함수를 시간의 존재 집계함수(time-dependent aggregate)라 한다. 또한 처리 형태에 따른 분류에서는 timefirst, first 등과 같이 특정 값을 찾아내는 선정 집계함수와 time, avg, rising 등과 같이 계산에 의해 결과를 산출하는 계산 집계함수로 분류한다[19].

논리적인 측면의 분류에서 처리 범위에 따른 분류는 각 시간 t에 대해 그 값이 시간 t에 유효한 튜플들을 대상으로 하는 즉각 집계함수와, 각 시간 t에 대해 그 값이 시간 t와 이전의 특정한

시간 간격상에서 유효한 튜플들을 대상으로 하는 누적 집계함수로 구분한다. 처리시점에 따른 분류는 주어진 특정한 캘린더 연도 동안에 유효한 모든 튜플을 대상으로 하는 정적 집계함수와 지정된 튜플이 가지는 시작시간 또는 이전에 유효한 모든 튜플을 대상으로 하는 동적 집계함수로 분류한다. 마지막으로 반환되는 형태에 의한 분류는 결과가 하나의 튜플을 반환하는 스칼라형

| 분류 기준 | 명 칭 | |
|------------|---------------|-----------------|
| .질의에사용된횟수 | 단일(single) | 중첩(nested) |
| .대상 속성의 종류 | 의존(dependent) | 독립(independent) |
| 처리 형태 | 선정(select) | 계산(compute) |

(a) 물리적인 측면의 분류

| 분류 기준 | 명 칭 | |
|-------|-------------------|----------------|
| .처리범위 | 즉각(instantaneous) | 누적(cumulative) |
| .처리시점 | 동적(dynamic) | 정적(static) |
| .반환유형 | 스칼라(scalar) | 릴레이션(relation) |

(b) 논리적인 측면의 분류

<그림 5> 집계함수의 분류

집계함수와 릴레이션을 반환하는 릴레이션형 집계함수가 있다[23].

한편 시간지원 집계함수 모델에서 사용하는 집계함수에는 기존의 집계함수인 sum, count, any, avg, min, max에 시간 개념을 추가하여 사용하는 것과 first, last, earliest, latest, stdev, avgti, vats 등 새로 정의된 집계함수가 있다[13][19].

다음은 시간지원 집계함수에 대한 정의이며, 집계함수 정의에 사용된 기호로서는 R은 릴레이션틀, r은 차수, n은 튜플의 개수 (count, any는 $n \geq 0$, sum, avg, min, max는 $n > 0$, 나머지는 $n > 2$), t는 R에 관련된 튜플 변수, s는 정렬되어진 릴레이션을 의미한다.

4.2.1 시간 개념이 추가된 기존의 집계함수

이 절에서 설명하는 count, any, min, max 등은 기존의 함수이름과 동일하더라도 시간대에 따라 집계기능을 수행하므로 내부 수행기능이 다르다. 이들에 대한 내부 수행은 [23]을 참조하며 본 논문의 연구에서 다루지 않는다

· count : 주어진 조건을 만족하는 튜플의 개수를 계산한다.

$$\text{count}(R) \triangleq (n, \dots, n)$$

· any : 주어진 조건을 만족하는 튜플이 존재하는지의 여부를 판단한다. 한 개의 튜플이라도 있을 경우 1, 그렇지 않을 경우 0을 반환한다.

$$\text{any}(R) \triangleq (\text{sign}(n), \dots, \text{sign}(n))$$

· sum : 주어진 조건을 만족하는 속성의 값을 더한다. 수치 속성만 사용할 수 있다.

$$\text{sum}(R) \triangleq \left[\sum_{t \in R} f[1], \dots, \sum_{t \in R} f[r] \right]$$

· avg : 주어진 조건을 만족하는 속성 값의 평균을 구한다. 수치 속성만 사용할 수 있다.

$$\text{avg}(R) \triangleq \left[\frac{1}{n} \sum_{t \in R} f[1], \dots, \frac{1}{n} \sum_{t \in R} f[r] \right]$$

· min : 주어진 조건을 만족하는 속성의 가장 작은 값을 구한다.

$$\text{min}(R) \triangleq (\min_{t \in R} f[1], \dots, \min_{t \in R} f[r])$$

· max : 주어진 조건을 만족하는 속성의 가장 큰 값을 구한다.

$$\text{max}(R) \triangleq (\max_{t \in R} f[1], \dots, \max_{t \in R} f[r])$$

4.2.2 새로운 시간지원 집계함수

새로운 집계함수는 앞절에서 기술한 집계함수 외에 새로운 기능을 추가한 것으로 first, last, earliest, latest, stdev 및 avgti 등이 있으며, 그 기능과 형식 표현은 다음과 같다.

· first : 각 튜플의 유효 시간 간격내에서 가장 오래된 유효 시작시간을 가진 속성의 값을 구한다. 두 개의 튜플이 동일한 유효시작 시간 값을

가지면 하나를 선택한다.

$$\text{first}(R) \triangleq t_{\text{first}}$$

$$R(t_{\text{first}}) \wedge (\forall t)(R(t) \wedge t \neq t_{\text{first}} \Rightarrow \\ \text{Before}(t_{\text{first}}[r+1], t[r+1]) \\ \vee \text{Equal}(t_{\text{first}}[r+1], t[r+1]))$$

- last : 각 튜플의 유효시간 간격 내에서 가장 최근의 유효 시작시간을 가진 속성 값을 구한다. 두 개의 튜플이 동일한 유효 시작시간 값을 가지면 임의로 하나를 선택한다.

$$\text{last}(R) \triangleq t_{\text{last}}$$

$$R(t_{\text{last}}) \wedge (\forall t)(R(t) \wedge t \neq t_{\text{last}} \Rightarrow \\ \text{Before}(t[r+1], t_{\text{last}}[r+1]) \\ \vee \text{Equal}(t[r+1], t_{\text{last}}[r+1]))$$

- earliest : 릴레이션에서 가장 오래된 유효시간을 갖는 튜플에서 그 시간 간격을 선택한다. 두 개의 튜플이 동일한 유효 시작시간 값을 가지면, 유효 종료시간이 더 오래된 시간 간격을 선택한다.

$$\text{earliest}(R) \triangleq [t_{\text{earliest}}[\text{from}], t_{\text{earliest}}[\text{to}]]$$

$$R(t_{\text{earliest}}) \wedge (\forall t)(R(t) \wedge t \neq t_{\text{earliest}} \Rightarrow \\ \text{Before}(t_{\text{earliest}}[r+1], t[r+1]) \\ \vee (\text{Equal}(t_{\text{earliest}}[r+1], t[r+1]) \\ \wedge (\text{Before}(t_{\text{earliest}}[r+2], t[r+2]) \\ \vee (\text{Equal}(t_{\text{earliest}}[r+2], t[r+2])))))$$

- latest : 릴레이션에서 가장 최근의 유효시간을 가지는 튜플에서 그 시간 간격을 선택한다. 두 개의 튜플이 동일한 유효 시작시간 값을 가지면 유효 종료시간이 더 최근의 시간 간격을 선택한다.

$$\text{latest}(R) \triangleq [t_{\text{latest}}[\text{from}], t_{\text{latest}}[\text{to}]]$$

$$R(t_{\text{latest}}) \wedge (\forall t)(R(t) \wedge t \neq t_{\text{latest}} \Rightarrow \\ \text{Before}(t[r+1], t_{\text{latest}}[r+1]) \\ \vee (\text{Equal}(t[r+1], t_{\text{latest}}[r+1]) \\ \wedge (\text{Before}(t[r+2], t_{\text{latest}}[r+2]) \\ \vee (\text{Equal}(t[r+2], t_{\text{latest}}[r+2])))))$$

- stdev(STandard DEVIation) : 주어진 속성 하에서 나타나는 n개의 값들에 대한 표준편차를 구한다. 이 집계함수는 수치 속성에 대해

서만 적용할 수 있다.

- avgti(AVeraGe Time Increment rate) : 시간에 따른 특정 속성 값에 대한 평균 증가율 또는 평균 감소율을 계산한다. 이 집계함수는 단일시점 릴레이션에서 수치 속성에 대해서만 적용할 수 있다. 시간적으로 이전 튜플의 속성값 사이에 대한 비교가 이루어지므로 적어도 두 개 이상의 튜플이 필요하고 두 개보다 적으면 avgti 결과 값은 0이 된다.

5. 결 론

시간지원 데이터 모델은 시간에 따라 변화되는 자료를 처리할 수 있도록 시간 의미를 데이터 모델에 추가하게 된다. 시간지원 데이터 모델은 지원되는 시간에 따라 유효시간 데이터 모델, 거래시간 데이터 모델, 그리고 이원시간 데이터 모델이 있다. 유효시간 데이터 모델은 실세계에서 사건이 발생한 시간을 지원하는 모델로 이 모델에서 사용하는 유효시간은 시점으로 표현하는 방법과 간격으로 표현하는 방법이 있으며 이들 시간을 튜플 단위로 결합하는 방법과 속성 단위로 결합하는 방법이 있다. 거래시간 데이터 모델은 사건이 데이터베이스에 수록된 시간을 지원하는 모델로 유효시간 데이터 모델과 유사한 방법에 의해 데이터 모델을 표현하며, 이 모델에서 처리하는 시간은 시스템에 의해서 처리되는 시간이므로 비삭제 정책을 사용해야 되는 응용 분야에서 사용할 수 있다. 이원시간 데이터 모델은 거래시간과 유효시간을 모두 지원하는 데이터 모델이 된다.

본 논문에서는 시간지원 데이터 모델의 시간 의미와 데이터 모델 구축시 고려사항을 살펴보고, 기존의 데이터 모델을 유효시간과 거래시간, 속성에 대한 동형성과 결합의 표현 여부에 따라 비교하였다. 한편 시간지원 데이터 모델을 설계할 경우에는 첫째, 데이터와 결합할 시간의 표현을 점 표현으로 할 것인지 간격으로 할 것

인지, 둘째, 시간의 결합을 튜플 단위로 할 것인지 속성 단위로 할 것인지, 셋째, 지원되는 시간이 유효시간인지 거래시간인지 아니면 두시간을 모두 지원할 것인지를 결정해야 한다. 그리고, 데이터에 시간 결합을 하므로서 파생되는 문제들, 즉 시간 연산, 시간 동형성, 시간의 일치성 등과 같은 문제들을 처리하기 위한 방법을 제시하여야 한다.

시간지원 데이터 모델에서 다루어야 할 문제 중의 하나는 시간지원에 따른 집계함수를 확장해야 한다. 그러므로 본 논문에서는 유효시간을

지원하고 시점과 간격으로 시간을 표현하며 튜플 타임스탬프를 사용하는 시간지원 집계함수 모델을 제시하였다. 기존의 집계함수도 시간지원 기능을 추가하면 원하는 시간대에 통계기능이 제공되도록 해야 하며, 시간지원 데이터 모델에서만 지원되는 집계함수에는 first, last, earliest, latest 등의 함수들이다.

앞으로의 연구에서는 본 연구에서 제안한 시간지원 집계모델의 질의 처리기의 설계와 구현이 필요하며, 또한 이원시간 데이터 모델을 정립하고 그에 따른 집계함수 모델도 확장하여야 한다.

〈참고문헌〉

- [1] G. A. Ariav, "A Temporally Oriented Data Model," ACM Transactions on Database Systems, 11, No. 4, pp. 499-527, Dec. 1986.
- [2] J. Ben-Zvi, "The Time Relational Model," PhD thesis, University of California at Los Angeles, 1982.
- [3] J. Clifford, A.U. Tansel, "On an Algebra for Historical Relational Databases: Two Views," in Proc. of ACM SIGMOD, pp. 247-265, May 1985.
- [4] J. Clifford, A. Croker, "The Historical Relational Data Model and Algebra Based on Lifespans," in Proc. of the International Conference on Data Engineering, IEEE Computer Society, pp. 528-537, Feb. 1987.
- [5] J. Clifford, Tomas Isakowitz, "On The Semantics of Transaction Time and Valid Time in Bitemporal Databases," Proceedings of International Workshop on an Infrastructure for Temporal Databases, June 1993.
- [6] S.K. Gadia, C.S. Yeung, "A Generalized Model for a Relational Temporal Database," in Proc. of ACM SIGMOD, pp. 251-259, JUMay 1985.
- [7] S. K. Gadia, "A Seamless Generic Extension of SQL for Querying Temporal Data," TR-92-02, Computer Science Department, Iowa State University, May 1992.
- [8] Christian S. Jensen, Michael D. Soo, Richard T. Snodgrass, "Unification of Temporal data Models," TR 92-15, Dept. of Mathematics and Computer Science, Univ. of Aalborg, July 1992.
- [9] Christian S. Jensen, Richard Snodgrass, "Proposal for a Data Model for the Temporal Structured Query Language," TR No. 37, Dept. of Computer Science, Univ. of Arizona, July 1992.
- [10] Christian S. Jensen, J. Clifford, S.K. Gadia, A. Segev, R.T. Snodgrass, "A Consensus Glossary of Temporal Database Concepts," SIGMOD Record, 23 No. 1, pp. 52-64, Mar. 1994.
- [11] S. Jones, P. Manson, R. Stamper, "LEGOL 2.0: A Relational Specification Language for Complete Rules," Information Systems, 4, No. 4, pp. 293-305, Nov. 1979.

- [12] D. H. Kim, K. H. Jeon, K. J. Jeong, K. J. Kim, K. H. Ryu, "A Temporal Database Management Main Memory Prototype," IEEE Region 10's Ninth Annual International Conference, pp. 391-396, Aug. 1994.
- [13] E. Mckenzie, R. Snodgrass, "Supporting Valid Time in a Historical Relational Algebra : Proofs and Extensions," TR 91-15, Dept. of Computer Science, Univ. of Arizona, Aug. 1991.
- [14] S. B. Navathe, R. Ahmed, "A Temporal Relational Model and a Query Language," Information Sciences, 49, pp. 147-175, 1989.
- [15] N. L. Sarda, "Extensions to SQL for Historical Databases," IEEE Transactions on Knowledge and Data Engineering, 2, No 2, pp. 220-230, June 1990.
- [16] N. L. Sarda, "HSQL : A Historical Query Language," Temporal Databases, Benjamin/Cummings, pp. 110-140, 1993.
- [17] A. Segev, A. Shoshani, "Logical Modeling of Temporal Data," in Proc. of ACM SIGMOD, pp. 454-466, May 1987.
- [18] R. T. Snodgrass, "The Temporal Query Language TQuel," ACM Transactions on Database Systems, 12, No 2, pp. 247-298, June 1987.
- [19] R. Snodgrass, S. Gomez, E. Mckenzie, "Aggregates in the temporal query language TQuel," TR-89-26, Dept. of Computer Science, Univ. of Arizona, Nov. 1989.
- [20] R. Snodgrass, Nick Kline, "Aggregate in TSQL2," The TSQL2 Language Design Committee, Mar. 21, 1994.
- [21] A. Tansel, "A Generalized Relational Framework for Modeling Temporal Data," Temporal Databases, Benjamin/Cummings, pp. 183-201, 1993.
- [22] Tansel, Clifford, Gadia, Jajodia, Segev, and Snodgrass, "Temporal Databases," The Benjamin/Cummings Publishing Company, Inc, 1993.
- [23] 김동호, 이인홍, 류근호, "주기억장치에서 시간지원 데이터베이스의 집계 함수 설계 및 구현," 한국정보과학회 논문지, 21권, 8호, pp. 1405-1415, 1994.
- [24] 김동호, 정경자, 전근환, 김기중, 류근호, "시간지원 데이터 관리 시험대," 한국정보처리 학회 논문지, 1권 1호, pp. 1-11, 1994년.
- [25] 남광우, 서경란, 류근호, "시간에 대한 연구," 충북대학교 데이터베이스 연구실, 1995.
- [26] 정경자, 전근환, 류근호, "시간지원 데이터베이스에서 거래시간 지원을 위한 거래시간 로그와 연산자의 설계 및 구현," 한국정보과학회 논문지, 22권 6호, PP. 849-859, 1995.