시간 순위 질의의 처리

권준호°, 송병호**, 이석호*
°*서울대학교 전기·컴퓨터공학부
**상명대학교 소프트웨어학과
bluerain@db.snu.ac.kr, bhsong@sangmyung.ac.kr, shlee@cse.snu.ac.kr

Temporal Ranked Query Processing

Joonho Kwon°, Byoungho Song**, Sukho Lee*
°*School of Electrical Engineering and Computer Science, Seoul National University
**Dept. of Software Science, Sangmyung University

요 약

시간에 따라 변화하는 사건을 기록하는 시간 데이타베이스에서는 사건을 저장할 때 시간 속성도 같이 저장한다. 최근에는 시간 데이타베이스의 속성을 고려하여 집계 함수와 같이 기존의 연산자를 확장하여 시간 데이타베이스에서 효율적으로 처리하려는 연구가 활발하게 진행되어 왔다.
사용자들은 종종 여러 애트리뷰트에 가중치를 두고 그 가중치 순서대로 결과가 보여지는 순위 질의를 실행한다. 기존의 순위 질의 개념을 그대로 시간 지원 데이타베이스에서 사용할 수 없다. 따라서 본 논문에서는 기존의 순위 질의에 시간 개념을 확장한 시간 순위 질의를 정의한다. 또한 시간 순위 질의 처리방법을 제시한다.

1. 서론

기존의 데이타베이스 시스템은 현실 세계에서 발생한 사건에 대하여 가장 최근의 상태만을 반영한다[1]. 새로운 데이타 값이 데이타베이스에 반영될 때, 새로운 값으로 기존의 데이타 값을 덮어쓰게 된다. 따라서 기존의 데이타베이스는 항상 가장 최근의 데이타를 저장하고 있다. 이에 비하여 시간 데이타베이스는 시간에 따라 변화하는 사건(event)이나 정보에 대한 질의와 저장을 지원한다. 사건들이 시간 데이타베이스에 저장될 때, 그 사건들의 유효 시간을 나타내는 시작 시간과 종료 시간 속성이 추가된다. 시간 데이타베이스 분야의 초기 연구는 주로 시간 데이타의 모델링 기법과 질의 언어 등과 같은 개념적인 문제들을 다루었으며, 색인 기법과 질의 처리 방법, 저장 구조 등과 같은 구현과 관련된 문제도 많은 연구가 수행되어 왔다. 최근에는 시간 개념의 추가로 인해 의미가 확장되는 연산자인 시간 집계 함수(temporal aggregation)에 관한 연구가 수행되어 왔다[2][3][4].

사용자들은 종종 여러 애트리뷰트에 가중치를 두고 그 가중치 순서대로 결과를 얻는 것을 원한다. 이 문제는 일상 생활에서 흔히 접하게 되는 문제로서, 예를 들면 집을 구할 때 사무실과의 거리, 가격, 면적 등과 같은 속성들을 고려하여 구매를 하는 경우를 들 수 있다. 본 논문에서는 이 연산자를 확장하여 시간 데이타베이스에서의 순위 질의(ranked query) 연산자를 설명하고 그 처리 기법도 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 순위 질의에 대한 관련 연구를 살펴보고 3장에서는 시간 순위 질의를 정의하고 그 처리 방법을 제안한다. 마지막으로 4장에서 결론을 맺는다.

2. 관련연구

여러 애트리뷰트들의 중요성에 따라 순서를 매겨서 데이타를 선택하는 문제를 효율적으로 처리하려는 여러 연구가 진행되었다.
[5]에서는 Onion Technique이라는 기법을 제시하였다. 이 기법은 관심 있는 점들은 투플 공간상의 convex hull

에 높이게 된다는 점을 이용하였다.

[6]에서는 순위 질의를 효율적으로 처리하기 위해서 가중치 함수의 결과값을 미리 저장해 둔 순위 뷰(ranked view)를 이용하는 기법을 제시하였다. 가중치 함수의 결과를 저장하는 순위 뷰를 적은 개수만 이용하여도 그 릴레이션에 나타날 수 있는 많은 개수의 가중치 함수들을 처리할 수 있음을 보이고, 이를 이용하여 PREFER라는 시스템을 구현하였다.

기존의 이런 방법들은 릴레이션에 속한 모든 데이타들에 대한 계산을 필요로 한다. 그러나 시간 데이타베이스에서는 사용자가 원하는 매 시간 구간마다 해당되는 투플들이 각각 다를 수 있기 때문에 위의 방법들을 그대로 사용할 수 없다.

3. 시간 순위 질의(Temporal Ranked Query)

3.1 시간 순위 질의의 정의

이 논문에서는 시간 데이타베이스에 기록되는 데이타들은 시작 시간 t_s, 종료 시간 t_e를 시간 속성으로 가지고 있고, 이 구간 $[t_s, t_e)$ 사이에서 이 데이타는 현실 세계에서 참이라고 가정한다. 즉 릴레이션 R은 m개의 속성과 2개의 시간 속성 (A_1, ..., A_m, t_s, t_e)로 구성되고, 속성 A_i는 숫자 타입을 가진다. $A_i(t)$는 투플 t의 i번째 속성의 값을 의미한다.

사용자가 요구하는 질의 q는 릴레이션과 가중치 함수, 시간 구간, 반환되길 원하는 결과의 수를 포함한다. 가중치는 $\vec{v} = (v_1, \ldots, v_m)$의 벡터 형식으로 표현한다. 각 가중치 값 v_i는 [0,1]사이의 값으로 정규화 되었고 $\sum_{j=1}^{m} v_j = 1$ 이라 가정한다. 본 논문에서는 가중치 함수는 선형적인 $f(t) = \sum_{j=1}^{m} v_j A_j(t)$ 형태라고 가정한다. 가중치 함수 f(t)는 각각의 투플 $t \in R$ 에 대하여 숫자 값을 반환한다. 선형이 아닌 다른 형태의 가중치 함수더라도 각 투플에 대해 숫자 값을 구할 수 있다면 시간 순위 질의의 결과를 정의할 수 있다.

가중치 함수가 선형일 때의 시간 순위 질위를 SQL 형식으로 표현하면 다음과 같다.

```
SELECT TOP k ID, v1A1 + v2A2 +...+ vmAm AS score
FROM r
ORDER BY SCORE DESC
r.VALID OVERLAPS Period t1, t2;
```

"TOP k"절에서 k가 순위에 따라 반환되는 결과의 개수를 의미하며, "$v_1A_1+...+v_mA_m$ AS score" 절이 가중치 함수를 의미한다. "r"은 릴레이션을 뜻한다. "Period t_1, t_2"가 사용자가 명시하는 시간 구간이며 "r.VALID OVERLAPS"절은 릴레이션 r에 속한 투플 중에서 사용자와 명세한 시간 구간과 투플의 유효 시간이 겹치는 데이타만 선택하도록 한다.

시간 순위 질의의 결과는 사용자가 원하는 상위 몇 개의 결과 값과 사용자가 명시한 시간 구간에 포함되는 시간 구간으로 이루어진 시퀀스가 된다.

시간 구간을 명시하지 않으면 릴레이션에 있는 투플들로 만들 수 있는 모든 구간에 대해서 가중치 함수를 적용하여 결과를 계산한다. 시간 구간을 시점으로 명세하면 그 시점을 포함하고 있는 투플만으로 가중치 함수를 수행하여 반환한다.

3.2 시간 순위 질의의 처리

시간 순위 질의를 처리하는 단계는 다음과 같다.

단계 1	사용자가 요구한 시간 구간에서 불변 구간을 구함
단계 2	각 불변 구간마다 단계 2-5를 반복
단계 3	불변 구간에 해당되는 투플들을 찾음
단계 4	해당되는 투플들에 가중치 함수 적용
단계 5	사용자가 명시한 개수만큼 투플들을 반환

단계 1에서 각각의 구간마다 해당하는 투플이 변하지 않는 구간을 불변 구간(constant interval)[2]이라고 한다. 단계 1을 빨리 수행하기 위해서 시간 데이타베이스를 위한 인덱스[7]를 사용할 수 있다.

3.3 시간 순위 질의의 예

그림 1(a)와 같은 예제 릴레이션 R이 있다. 사용자가 가중치로 $\vec{v}$ =(0.1, 0.6. 0.3), 순위 질의의 결과를 알고 싶은 구간 [13,20), 그 구간에서 반환될 결과의 수 k=2

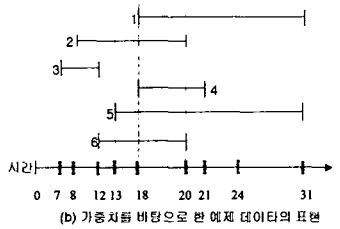
를 명세한 아래의 질의를 실행하였다고 하자.

```
SELECT TOP 2 ID, 0.1A1 + 0.6A2 + ... + 0.3Am AS score
FROM R
ORDER BY SCORE DESC
R.VALID OVERLAPS Period 13, 20;
```

예제 릴레이션의 데이타에 대해 가중치 함수를 계산한 것이 f(t) 필드이다. 이 가중치 함수 값을 바탕으로 데이타를 시간축에 표시하면 1(b)와 같다. 사용자가 명세한 구간 [13,20)에서 불변 구간을 구하면 [13,18)과 [18,20)이 된다. 각각의 불변 구간에 해당되는 투플을 찾으면 [13,18) 사이에 해당하는 투플의 ID는 2, 6, 5이며 [18,20) 사이의 구간에 해당되는 투플의 ID는 1, 2, 4, 6이다. 각 불변 구간에 해당되는 이들 투플만으로 가중치 함수를 계산하여 결과를 구한다. 따라서 사용자가 원하는 구간이 [13,20)의 순위 질의 결과는 1(c)와 같은 형태로 사용자에게 반환된다.

ID	A1	A2	A3	t_s	t_e	$F_w(d)$
1	10	17	20	18	31	17.2
2	20	20	11	8	20	17.3
3	17	18	12	7	12	16.1
4	15	10	8	18	21	9.9
5	5	10	12	13	31	10.1
6	15	10	5	12	20	5

(a) 예제 릴레이션 R



(b) 가중치를 바탕으로 한 예제 데이타의 표현

ID	t_s	t_e
2	13	18
5	13	18
1	18	20
2	18	20

(c) 결과

그림 1 시간 순위 질의의 예

4. 결론

본 논문에서는 시간 데이타베이스에 필요로 하는 연산자로 기존의 순위 질의를 확장한 시간 순위 질의를 정의하고 처리 방법을 제시하였다.

이 논문에서는 사용자가 제시하는 가중치는 변하지 않는 것으로 생각을 하였다. 그러나 시간에 따라서 가중치도 변할 수 있으므로, 시간 구간별로 가중치 함수를 명세하는 것을 고려한 시간 순위 질의에 대한 연구가 더 필요하다.

참고문헌

[1] C. S. Jensen and R. T. Snodgrass, "Temporal Data Management", IEEE TKDE 11(1), pp. 36-44, 1999

[2] N. Kline and R.T. Snodgrass, "Computing Temporal Aggregates", In Proc. of the 11th Inter. Conference on Data Engineering, pp. 222-231, 1995.

[3] Bongki Moon et al, "Scalable Algorithms for Large Temporal Aggregation", In Proc. of the 16th Inter. Conference on Data Engineering, pp. 145-154, 2000.

[4] Jun Yang, Jennifer Widom, "Incremental Computation and Maintenance of Temporal Aggregates", In Proc. of the 17th Inter. Conference on Data Engineering, pp. 51-60, 2001

[5] Y. chi Chang et al. "The Onion Technique: Indexing for Linear Optimization Queries", ACM SIGMOD, pp. 391-402, 2000.

[6] Vagelis Hristidis et al. "PREFER: A System for the Efficient Execution of Multi-parametric Ranked Queries", ACM SIGMOD, pp , 2001

[7] B. Salzberg and, V.J. Tsotras, "A Comparison of Access Methods for Temporal Data", ACM Computing Surveys 31(2), 1999.