효율적인 XML 질의 처리를 위한 XQuery 질의의 정규화

김서영 ○† 이기훈† 황규영†

†한국과학기술원 전산학과/첨단정보기술연구센터

{sykim, drlee, kywhang}@mozart.kaist.ac.kr

Normalization of XQuery Queries for Efficient XML Query Processing

Seo-Young Kim○† Ki-Hoon Lee† Kyu-Young Whang†

†Department of Computer Science &

Advanced Information Technology Research Center

Korea Advanced Institute of Science and Technology

요 약

XML 이 웹 상에서의 정보 표현, 통합, 교환을 위한 표준이 됨에 따라 다양한 XML 질의 언어들이 제안되었으며, World Wide Web Consortium(W3C)은 XQuery 를 XML 질의 언어의 표준으로 권고하였다. XQuery 는 SQL 과 유사하게 중첩 질의를 허용하므로, 중첩된 XQuery 질의를 동일한 의미를 가지면서 보다 효율적으로 실행될 수 있는 질의로 변환하는 정규화 규칙들이 제안되었다. 그러나 제안된 정규화 규칙들은 제한적인 형태의 중첩 질의에만 적용되는 문제점을 가지고 있다. 특히, FLWR 표현식의 where 절에 있는 중첩을 처리할 수 없다. 본 논문에서는 SQL 질의의 정규화 규칙들을 확장하여 FLWR 표현식의 모든 절에 나타나는 중첩을 처리할 수 있는 XQuery 질의의 정규화 규칙들을 제안한다. 이를 위해 먼저, 상관과 집계의 유무에 따라 XQuery 질의의 중첩 유형을 분류하고, 각 유형 별로 정규화 규칙들을 제안한다. 다음으로, 중첩된 XQuery 질의에 정규화 규칙들을 적용하는 세부 알고리즘을 제안한다.

1. 서 론

XML(eXtensible Markup Language) 문서는 구조 정보를 추가할 수 있는 문서로서, 웹의 새로운 표준 문서로 많이 사용되고 있다[1]. 이에 따라, 대량의 XML 문서들에 대해 효율적으로 질의할 수 있는 시스템의 필요성이 점차 부각되었으며, Quilt, XQL, XML-QL, XPath, XQuery 등과 같은 다양한 XML 질의 언어들이 등장하였다. 이 중에서 XPath 와 XQuery 가 가장 널리 사용되고 있는 XML 질의 언어이다.

XPath 는 XML 문서를 엘리먼트와 애트리뷰트들로 이루어진 트리로 모델링하고, 트리 상에서 노드가 되는 엘리먼트와 애트리뷰트들을 검색할 수 있는 표준 질의 언어이다[2]. XQuery 는 XPath 를 포함하는 질의 언어로서 현재 *World Wide Web Consortium*(W3C)에서 표준화가 진행 중이다. XQuery 는 XPath 를 이용하여 XML 문서의 엘리먼트와 애트리뷰트들에 접근(access)할 수 있을 뿐만 아니라, 여러 XML 문서들을 조인하거나 새로운 XML 문서를 생성할 수 있는 기능을 제공한다[3]. 특히 XQuery 는 SQL 의 select-from-where 문과 유사한 기능을 가지는 FLWR 표현식을 제공한다. FLWR 표현식은 for 절, let 절, where 절, 그리고 return 절로 구성되며, 각 절 안에 또 다른 FLWR 표현식이 중첩될 수 있는 특징이 있다.

그러나 중첩된 FLWR 표현식을 포함하는 XQuery 질의는 중첩된 루프문의 실행을 야기하기 때문에, 실제 시스템에서 처리될 때는 비효율적인 문제가 있다. 따라서 중첩된 XQuery 질의를 동일한 의미를 가지면서 보다 효율적으로 실행될 수 있는 질의로 변환하는 정규화 규칙들이 필요하다.

XQuery 질의의 정규화에 관한 연구로는 참고 문헌 [4] 등의 연구가 있다. 참고 문헌 [4]에서는 복잡한 XQuery 질의를 단순화하고, 중첩이 있는 XQuery 질의에서 중첩을 제거하는 규칙들을 제안하였다. 그러나 제안된 XQuery 질의의 정규화 규칙들 중에서 질의를 구성하는 주요 표현식인 FLWR 표현식의 중첩을 제거하는 규칙은 매우 제한적이라는 문제점이 있다. 즉, FLWR 표현식의 where 절에 중첩이 있는 경우에 대해 고려하기 않았으며, for 절과 return 절에 중첩이 있는 경우에도 매우 제한적인 형태에 대해서만 정규화 규칙을 제안하였다.

본 논문에서는 SQL 질의의 정규화 규칙들을 확장하여 XQuery 질의를 구성하는 FLWR 표현식의 모든 절에 나타나는 중첩을 제거하는 정규화 규칙들을 제안한다. 본 논문의 주요 공헌은 다음과 같다. 첫째, 상관과 집계의 유무에 따라 XQuery 질의의 중첩 유형을 분류하고, 각 유형 별로 정규화 규칙을 제안한다. 둘째, 중첩된 XQuery 질의에 정규화 규칙들을 적용하는 세부 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 관련 연구로서 기존의 XQuery 질의의 정규화와 SQL 질의의 정규화에 대해 설명한다. 제 3 장에서는 확장된 XQuery 질의의 정규화 규칙들과 이를 적용하는 세부 알고리즘에 대해 설명한다. 마지막으로 제 4 장에서는 결론을 내린다.

2. 관련 연구

본 장에서는 관련 연구로서 XQuery 질의의 정규화와 SQL 질의의 정규화에 대해서 설명한다. 제 2.1 절에서는 XQuery 질의의 정규화에 대해 설명하고, 제 2.2 절에서는 SQL 질의의 정규화에 대해 설명한다.

2.1. XQuery 질의의 정규화

실세계에서 대부분의 데이터는 관계형 데이터 형태로 저장되어 있으나, XML 문서가 정보 교환의 표준으로 많이 사용됨에 따라 관계형 데이터를 XML 문서 형태로 접근(access)하고자 하는 요구가 발생하였다. 이를 위해 기존의 관계형 데이터를 가상의 XML 문서로 표현하고, 이 가상의 XML 문서에 대해 XQuery 질의를 수행하고자 하는 연구[4]가 이루어졌다. 참고 문헌 [4]에서는 XQuery 질의를 SQL 질의로 변환하기 쉬운 형태로 바꾸기 위해, XQuery 질의를 단순화하고 질의의 중첩을 제거하는 정규화 규칙들을 제안하였다.

그러나 이 논문에서는 제한적인 형태의 중첩 유형에 대해서만 정규화 규칙들을 제안하였다. 즉, SQL 질의에서는 where 절의 중첩이 허용되므로 FLWR 표현식의 where 절에 중첩이 있는 경우에 대해서는 정규화 규칙을 제안하지 않았고, for 절과 return 절에 중첩이 있는 경우에 대해서만 정규화 규칙을 제안하였다. 그런데 for 절과 return 절에 중첩이 있는 경우 중에서도 FLWR 표현식이 집계 함수의 인자로 중첩된 경우에 대해서는 정규화 규칙을 제안하지 않았다.

2.2. SQL 질의의 정규화

SQL(Structured Query Language)은 현재 가장 널리 쓰이고 있는 표준 관계형 데이터베이스 질의 언어로서 질의를 중첩하여 표현할 수 있는 특징이 있다. 그러나 데이터베이스 시스템에서 중첩된 질의를 처리하는 것은 중첩된 루프문의 실행을 야기하기 때문에 비효율적이다. 따라서 중첩된 질의를 중첩되지 않은 질의로 변환하여 처리함으로써 질의 처

* 본 연구는 첨단정보기술연구센터를 통하여 한국과학재단의 지원을 받았음.

리 성능을 향상시키고자 하는 연구([5], [6], [7])가 진행되었다. SQL 질의의 기본 구조는 select 절, from 절, where 절로 구성된 질의 블록으로 표현할 수 있다. 하나의 질의 블록 안에는 또 다른 질의 블록이 중첩될 수 있는데, 이때 안쪽 질의 블록의 where 절에 바깥쪽 질의 블록의 릴레이션을 참조하는 부분이 있으면 상관(correlation)이 있는 것이며, 질의 블록의 select 절에 집계 함수가 있으면 집계(aggregation)가 있는 것이다. 참고 문헌 [5]에서는 SQL 질의에 나타나는 where 절의 중첩을 상관과 집계 유무에 따라 Type-A, Type-N, Type-J, Type-JA, 그리고 Type-D 유형으로 분류하고, 각 유형 별로 질의를 정규화하는 규칙을 제안하였다.

Type-A 유형의 중첩 질의는 안쪽 질의 블록에 상관은 없고 집계만 있는 경우로서, 안쪽 질의 블록을 미리 수행하여 얻은 결과 값을 안쪽 질의 블록과 치환함으로써 중첩을 제거한다. Type-N 유형의 중첩 질의는 안쪽 질의 블록에 상관과 집계가 모두 없는 경우이고, Type-J 유형의 중첩 질의는 안쪽 질의 블록에 상관은 있고 집계는 없는 경우이다. Type-N 유형과 Type-J 유형의 중첩 질의는 바깥쪽 질의 블록의 릴레이션과 안쪽 질의 블록의 릴레이션을 조인함으로써 중첩을 제거한다. Type-JA 유형의 중첩 질의는 안쪽 질의 블록에 상관과 집계가 모두 있는 경우로서, 안쪽 질의 블록에서 각 상관 값에 대한 집계 함수의 값들을 미리 계산하여 임시 릴레이션으로 생성하고 이 임시 릴레이션을 바깥쪽 질의 블록의 릴레이션과 조인함으로써 중첩을 제거한다. 마지막으로 Type-D 유형의 중첩 질의는 바깥쪽 질의 블록의 where 절에 두 개의 질의 블록을 피연산자로 갖는 연산이 있고 이 중 하나 이상의 안쪽 질의 블록에 상관이 있는 경우로서, 디비전(division) 연산을 이용하여 처리한다.

3. XQuery 질의의 정규화 규칙

본 장에서는 확장된 XQuery 질의의 정규화 규칙들에 대해 설명한다. 제 3.1 절에서는 중첩된 XQuery 질의의 유형을 분류하고, 제 3.2 절에서는 중첩 유형에 따라 XQuery 질의를 정규화하는 규칙들에 대해 설명한다. 제 3.3 절에서는 XQuery 질의의 정규화 규칙들을 적용하는 세부 알고리즘에 대해 설명한다.

3.1. 중첩 질의의 유형 분류

XQuery 질의는 SQL 질의와 마찬가지로 상관과 집계의 유무에 따라 중첩된 질의의 유형을 분류할 수 있으며, XQuery 질의에서의 상관과 집계는 SQL 질의에서와 유사한 의미를 가진다. 상관과 집계의 유무에 따라 XQuery 의 FLWR 표현식에서 가능한 중첩의 유형을 분류하면 표 1 과 같다. 표에서 흰색 원으로 표시된 칸은 기존 연구 [4]에서 제안된 정규화 규칙이 있는 경우이다. 그리고 검은색 원으로 표시된 칸은 기존 연구 [4]에서 정규화 규칙을 제안하지 않았기 때문에 본 논문에서 SQL 질의의 정규화 규칙을 확장하여 새로운 정규화 규칙을 제안한 경우이다. 특히, where 절에 Type-D 유형의 중첩이 있는 경우에 SQL 질의에서는 where 절 연산이 두 안쪽 질의 블록의 질의 결과 간의 포함 관계를 묻는 연산을 의미하는데 비해, XQuery 질의에서는 두 안쪽 FLWR 표현식의 결과 간의 비교 연산을 의미한다. 따라서 SQL 질의의 정규화 규칙과는 다른 새로운 정규화 규칙을 필요하다.

유형 \ 절	for	where	return
Type-A	●	●	●
Type-N	○	●	○
Type-J	○	●	○
Type-JA	●	●	●
Type-D	N/A	●	N/A

○: 기존 연구 [5]에서 제안한 정규화 규칙
●: 본 논문에서 제안한 정규화 규칙

표 1. FLWR 표현식에서 가능한 중첩 유형의 분류

3.2. 정규화 규칙

FLWR 표현식의 각 절에 Type-A 유형과 Type-JA 유형의 중첩이 있는 경우에는 참고 문헌 [5]에서 제안한 정규화 규칙을 XQuery 문법에 맞게 확장하여 적용한다. where 절에 Type-N 유형과 Type-J 유형의 중첩이 있는 경우에는 참고 문헌 [5]의 정규화 규칙을 XQuery 문법에 맞게 확장하여 그대로 적용하면 안쪽 질의 블록의 질의 결과에 중복된 값이 존재하는 경우에 정규화하기 전과 후의 질의 결과가 달라지는 문제점이 있으므로 수정하여 적용한다. where 절에 Type-D 유형의 중첩이 있는 경우에는 참고 문헌 [5]에서 정의한 SQL 의 Type-D 유형 중첩과 그 의미가 다르기 때문에 새로운 정규화 규칙을 제안한다.

먼저 where 절에 Type-N 유형과 Type-J 유형의 중첩이 있는 경우를 처리하는 수정된 정규화 규칙에 대해 설명하고, 다음으로 Type-D 유형의 중첩이 있는 경우를 처리하는 새로운 정규화 규칙에 대해 설명한다.

■ Type-N 유형과 Type-J 유형의 중첩 질의에 대한 정규화 규칙

Type-N 유형과 Type-J 유형의 중첩 질의에 대한 SQL 의 정규화 규칙은 안쪽 질의 블록의 질의 결과에 중복된 값이 존재하는 경우에 정규화하기 전과 후의 질의 결과가 달라지는 문제점이 있다. 예를 들어 Type-J 유형의 중첩 질의인 질의 1 에 대해 참고 문헌 [5]에서 제안한 정규화 규칙을 그대로 적용하면, 안쪽 질의 블록의 결과에 중복된 값이 없다고 가정하고 조인 형태로 변환하여 질의 1-1 과 같이 정규화한다. 그러나 안쪽 FLWR 표현식의 결과에 $x/PNO 와 같은 값의 $y/PNO 가 두 개 이상 존재하는 경우 질의 1 에서는 $x/PName 이 한번만 반환되는데 반해, 질의 1-1 에서는 같은 값을 가지는 $y/PNO 의 개수만큼 반환된다.

질의 1

```
for    $x in document("Projs.xml")/Projs/Proj
where $x/PNO
        =
       ( for    $y in document("WorksOn.xml")/WorksOnList/WorksOn
         where $y/WYear = $x/PYear
         return $y/PNO )
return $x/PName
```

질의 1-1

```
for    $x in document("Projs.xml")/Projs/Proj,
       $y in document("WorksOn.xml")/WorksOnList/WorksOn
where $x/PNO = $y/PNO and $y/WYear = $x/PYear
return $x/PName
```

이러한 문제는 안쪽 FLWR 표현식을 먼저 수행하여 얻은 질의 결과에서 중복된 값들을 제거한 후, 그 결과를 주 FLWR 표현식과 조인하여 정규화함으로써 해결할 수 있다. 본 논문에서 제안한 정규화 규칙은 그림 1 과 같다. 그림에서 $1 은 $t 에 바인딩된 값의 쌍에서 첫번째 값을 의미하고, $2 는 두 번째 값을 의미한다. 질의 1 에 본 논문에서 제시한 정규화 규칙을 적용하면 질의 1-2 와 같다. Type-N 유형의 중첩 질의인 경우에는 그림 1 의 정규화 규칙에서 상관 값과 관련된 부분을 제거하고 적용하면 된다.

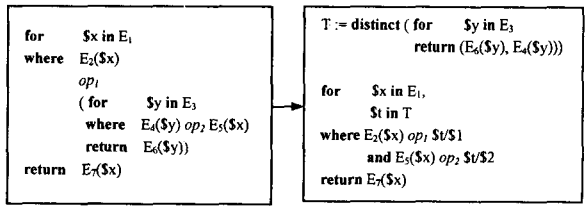

그림 1. Type-J 유형 중첩 질의의 정규화 규칙

질의 1-2

```
T := distinct (for    $y in document("WorksOn.xml")/WorksOnList/WorksOn
                return ($y/PNO, $y/WYear))
for    $x in document("Projs.xml")/Projs/Proj,
       $t in T
where $x/PNO = $t/$1 and $x/PYear = $t/$2
return $x/PName
```

■ Type-D 유형의 중첩 질의에 대한 정규화 규칙

XQuery 에서 Type-D 유형 중첩 질의는 주 FLWR 표현식의 where 절에 두 개의 안쪽 FLWR 표현식을 피연산자로 갖는 연산이 있고, 이 중에서 하나 이상의 안쪽 FLWR 표현식에 주 FLWR 표현식과의 상관이 있

는 경우를 말한다. 여기서 두 안쪽 FLWR 표현식 간에는 상관이 존재하지않기 때문에, 각 안쪽 FLWR 표현식을 독립적으로 수행하여 처리할 수 있다. 따라서 각 안쪽 FLWR 표현식의 중첩 유형에 따라 해당하는 정규화 규칙을 차례로 적용함으로써 정규화 할 수 있다. 본 논문에서 제안한 Type-D 유형 중첩 질의에 대한 정규화 규칙은 다음과 같다.

Type-D 유형 중첩 질의의 정규화 규칙

1. 중첩된 두 안쪽 FLWR 표현식 중에서 하나의 안쪽 FLWR 표현식이 주 FLWR 표현식에 대해 이루는 중첩 유형의 정규화 규칙에 따라 안쪽 FLWR 표현식을 미리 수행한다.
2. i 에서 정규화하지 않은 다른 안쪽 FLWR 표현식이 주 FLWR 표현식에 대해 이루는 중첩 유형의 정규화 규칙에 따라 안쪽 FLWR 표현식을 미리 수행한다.
3. 1 과 2 에서 미리 계산한 안쪽 FLWR 표현식의 결과들을 조인하고 상관 값과의 조인에 참여하는 값만 선택한 후, 중복을 제거한다.
4. 주 FLWR 표현식과 3 에서 생성된 결과를 조인한다.

정규화 규칙에서 중첩된 XQuery 질의를 세 개의 FLWR 표현식이 조인된 형태로 직접 변환하지 않고, 위와 같은 단계를 거치는 이유는 Type-N 과 Type-J 유형의 중첩 질의에서와 마찬가지로 중복된 값으로 인한 문제가 발생할 수 있기 때문이다.

3.3. 정규화 규칙들을 적용하는 세부 알고리즘

본 절에서는 확장된 XQuery 질의의 정규화 규칙들을 적용하기 위한 세부 알고리즘에 대해 설명한다. 먼저 정규화를 위한 자료 구조에 대해 설명하고, 다음으로 자료 구조 상에서의 정규화 알고리즘에 대해 설명한다.

■ 정규화를 위한 자료 구조

본 논문에서는 하나의 FLWR 표현식을 하나의 질의 상자로 표현하여, 안쪽 FLWR 표현식과 바깥쪽 FLWR 표현식 간의 상관을 알기 쉽게 표현해 주는 자료 구조를 제안하였다. 각 질의 상자는 헤드(head)와 바디(body)로 구성되는데, 먼저 헤드는 FLWR 표현식의 return 절에 대응되는 것으로서 바디에서 반환되어야 할 질의의 결과를 나타낸다. 그리고 바디는 FLWR 표현식의 for 절과 where 절에 대응되는 것으로서 for 절에 대응되는 노드(node)와 where 절에 대응되는 에지(edge)로 구성된다. 바디에서 동일한 노드에 대한 에지는 실렉션 조건을 의미하며, 서로 다른 노드 간의 에지는 조인 조건을 의미한다. 특히, 서로 다른 질의 상자에 속한 노드 간의 에지는 각 질의 상자가 나타내는 FLWR 표현식 간에 상관이 있음을 의미한다. 마지막으로 각 질의 상자는 distinct 를 나타내는 플래그와, group by 와 order by 에 대한 정보를 부가적으로 가질 수 있다.

예를 들어, Type-J 유형의 중첩 질의인 질의 1 을 정규화를 위한 자료 구조로 표현하면 그림 2 와 같다. 그림 2 에서 바깥쪽 FLWR 표현식은 질의 상자 QB_1 에 해당되고, 안쪽 FLWR 표현식은 질의 상자 QB_2 에 해당된다. 그리고 바깥쪽 FLWR 표현식의 where 절은 피연산자로 질의 상자 QB_2 를 갖는 노드 $x 의 실렉션 조건으로 표현되고, 안쪽 FLWR 표현식의 where 절은 노드 $x 와 노드 $y 간의 조인 조건으로 표현된다. 이렇게 서로 다른 질의 상자에 속하는 노드 간의 에지를 보고 FLWR 표현식 간의 상관을 쉽게 파악할 수 있다.

■ 자료 구조 상에서의 정규화 알고리즘

자료 구조상에 정규화 규칙들을 적용하는 알고리즘 Normalize_QG 는 그림 3 과 같다. Normalize_QG 는 입력 받은 질의 상자가 나타내는 FLWR 표현식의 for 절, where 절, return 절에 있는 중첩에 대해 차례로 정규화 규칙을 적용한다. 즉 먼저, 질의 상자의 노드에 있는 중첩을 제거하고, 다음으로 질의 상자의 에지가 나타내는 실렉션 조건에 있는 중첩을 제거한다. 마지막으로 질의 상자의 헤드에 있는 중첩을 제거한다.

4. 결 론

본 논문에서는 효율적인 XML 질의 처리를 위해 XQuery 질의를 정규화하는 방법을 제안하였다. 이를 위해, XQuery 질의를 구성하는 주요 표현식인 FLWR 표현식의 모든 절에 나타나는 중첩을 처리할 수 있는 정규화 규칙들을 제안하였으며 이를 적용하는 세부 알고리즘들을 제안하였다.

본 논문의 주요 공헌은 다음과 같다. 첫째, 상관과 집계의 유무에 따라 XQuery 질의의 중첩 유형을 분류하고, 각 유형 별로 정규화 규칙을 제안하였다. 둘째, 제안한 규칙들을 사용하여 중첩된 XQuery 질의를 정규화하는 세부 알고리즘을 제안하였다. 이를 위해 먼저 정규화를 위한 자료 구조를 제안하였으며, 자료 구조 상에서 제안한 정규화 규칙들을 적용하는 알고리즘을 제안하였다.

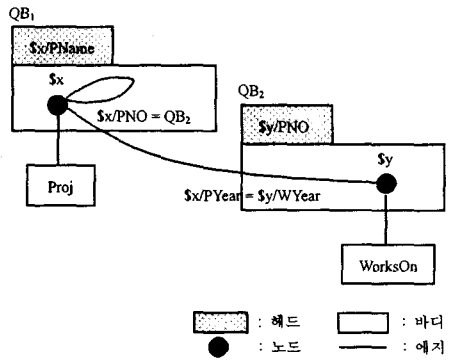

그림 2. 정규화를 위한 자료 구조의 예.

```
Normalize_QG (질의 상자 QB1) {

    /* for 절의 중첩을 제거 */
    if (QB1의 노드에 다른 질의 상자 QB2가 바인딩됨) {
        Normalize_QG(QB2);
        상관과 집계의 유무에 따라 해당하는 정규화 규칙 적용;
    }

    /* where 절의 중첩을 제거 */
    if (QB1의 실렉션 조건에 하나 이상의 다른 질의 상자가 포함됨) {
        if (피연산자 중에서 하나만 질의 상자 QB2임) {
            Normalize_QG(QB2);
            상관과 집계의 유무에 따라 해당하는 정규화 규칙 적용;
        }
        else if (피연산자가 두 개 모두 질의 상자이고 상관이 존재함)
            Type-D 유형의 중첩 질의를 정규화 하는 규칙 적용;
    }

    /* return 절의 중첩을 제거 */
    if (QB1의 헤드에 다른 질의 상자 QB2가 포함됨) {
        Normalize_QG(QB2);
        상관과 집계의 유무에 따라 해당하는 정규화 규칙 적용;
    }
}
```

그림 3. 자료 구조 상에서의 정규화 알고리즘.

참고 문헌

[1] Simon, H., *Strategic Analysis of XML for Web Application Development*, Computer Research Corp., 2000.

[2] World Wide Web Consortium, XML Path Language (XPath) Version 1.0, W3C Recommendation, Nov. 1999 (availiable from http://www.w3.org/TR/xpath.html).

[3] World Wide Web Consortium, XQuery 1.0: An XML Query Language, W3C Working Draft, Aug. 2003 (availiable from http://www.w3.org/TR/xquery/).

[4] Manolescu, I., Florescu, D., and Kossmann, D., "Answering XML Queries over Heterogeneous Data Sources," In *Proc. 27th Int'l Conf. on Very Large Data Bases*, Roma, Italy, pp. 241-250, Sept. 2001.

[5] Kim, W., "On Optimizing an SQL-like Nested Query," *ACM Trans. on Database Systems*, Vol. 7, No. 3, pp. 443-469, Sept. 1982.

[6] Ganski, R. and Wong, H., "Optimization of Nested SQL Queries Revisited," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, San Francisco, USA, pp. 23-33, May 1987.

[7] Seshadri, P., Pirahesh, H., and Leung, T., "Complex Query Decorrelation," In *Proc. the 17th Int'l Conf. on Data Engineering*, New Orleans, Louisiana, pp. 450-458, Feb. 1996.