사용자 관점에서의 조인 연산 평가 방법론

정 회 진[†] · 이 상 호[††]

요 약

조인 연산은 데이터베이스 시스템의 가장 기본적이고 중요한 연산이며, 큰 수행 비용을 필요로 하는 연산이다. 지금까지 효율적인 조인 연산의 구현과 평가를 위한 다양한 연구가 진행되었다. 기존의 조인 연산 평가는 구현자 관점에서 우수한 알고리즘을 규명하기 위해 개발되었다. 본 논문은 사용자 관점에서 조인 연산 성능을 평가하는 조인 연산 평가 방법론을 기술한다. 본 성능 평가는 사용자가 자신의 업무에 적합한 조인 연산을 수행하는 데이터베이스 시스템을 선택하는데 도움을 준다. 본 논문에서는 두 가지 조인 범주와 여섯 가지의 조인 성능 영향 요소를 고려하여 42개의 평가 질의를 제안한다. 마지막으로 제안된 방법론에 따라 두 개의 상용 데이터베이스 시스템을 대상으로 구현하고 성능 평가 결과를 기술하였다.

A Join Operations Benchmark in Users' Perspective

Hoe Jin Jeong[†] · Sang Ho Lee[††]

ABSTRACT

The join operation is an important, fundamental operation in database systems, and it costs much to execute. In the literature, there are a number of technical attempts on development and evaluation of efficient join operations, all of which have been carried out in developers' perspective. This paper proposes a join operations benchmark that is dedicated to the evaluation of the join operations in database systems in users' perspective. This benchmark helps users select a database system that performs the join operations well in their work environment. The benchmark consists of 42 join queries, which are derived from on six performance factors that are picked out in two join categories. We have implemented this benchmark with two commercial database systems. The experimental results are also reported.

키워드 : 성능 평가(Benchmark), 조인 연산(Join Operation), 사용자 관점(Users' Perspective), 조인 알고리즘(Join Algorithm)

1. 서 론

데이터베이스 시스템의 많은 연산 중에서 조인 연산은 정보간에 물리적으로는 전혀 연결 관계가 없을지라도 필요시 논리적인 관계만으로도 원하는 정보를 서로 연결하여 참조할 수 있도록 해주는 데이터베이스 시스템의 가장 기본적이고 중요한 기능 중 하나이다. 질의 처리에 있어 데이터 집중적이고 수행 시간이 오래 걸리는 연산이기 때문에 지금까지 효율적인 조인 연산에 대한 구현과 성능 평가에 대한 많은 연구가 있어 왔다.

새로운 조인 알고리즘을 개발하여 해당 알고리즘이 여타 다른 조인 알고리즘에 비해 개선되고 더 나은 성능 보인다는 방향[1, 2, 3, 4, 5]과 지금까지 개발된 다양한 조인 알고리즘들을 비교 분석[6, 7]하여 개발자들에게 좋은 알고리즘을 선택하도록 도움을 주는 방향에서의 연구 노력들은 모두 구현자의 관점에서 우수한 조인 알고리즘을 규명하는데 그 목적을 두고 있다. 여러 조인 연산을 어떻게 수행하게 할 것인지, 조인 알고리즘에서 어떤 부분을 개선하는 것이 성능 개선에 많은 영향을 미치는지 등 데이터베이스 시스템을 개발하는 입장에서 질의 최적기에 초점을 맞추어 관련 연구가 이루어졌고 그에 따른 성능 평가가 개발되어 수행되었다.

사용자 관점에서의 조인 연산 평가 방법론은 조인 연산을 빈번히 수행하는 사용자가 자신의 업무에서 사용되는 조인 연산을 가장 적합하게 수행하는 데이터베이스 시스템을 선택하기 위해 사용할 수 있는 성능 평가 방법론을 말하며, 사용자 관점에서의 조인 연산 평가 방법론은 현재까지 부재하다. 사용자가 데이터베이스 시스템 선택을 위해 기존 성능 평가 방법론들의 일부 조인 관련 질의만을 사용하여 조인 연산의 성능을 평가하는 것이 한가지 대안이 될 수 있다.

위스콘신 성능 평가[8], SetQuery 성능 평가[9], BORD 성능 평가[10]는 관계형 데이터베이스시스템이나 객체-관계형 데이터베이스 시스템의 질의 최적기 성능 측정을 목적으로 개발된 성능 평가들로서 모두 조인 질의를 포함하고 있

※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌습니다.
† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
†† 정 회 원 : 숭실대학교 컴퓨터학부
논문접수 : 2004년 8월 11일, 심사완료 : 2004년 11월 16일

다. 각 성능 평가의 조인 질의들은 동등 조인(equijoin)만을 사용하여 정의되었으며, 조인 대상이 되는 테이블의 개수도 2개 혹은 3개만을 사용하여 조인 연산의 전반적인 수행 능력을 평가하기에는 질의 범위가 부족하다. 산업계에서 온라인 트랜잭션 처리(online transaction processing) 분야나 웹 기반 전자 상거래 분야, 의사 결정 지원 시스템(decision supporting system) 분야 등에서 성능 평가를 위해 널리 사용되고 있는 TPC 성능 평가들[11]도 모두 동등 조인 연산만을 대상으로 하고 있다는 단점을 가진다. TPC 성능 평가들은 조인 대상 테이블로 성능 평가 종류에 따라 다소 차이가 있긴 하지만 최소 2개에서 최대 8개까지를 사용하고 있다. 하지만 조인 대상 테이블이 조인 연산마다 달라 그 결과를 통한 가치 있는 통계 정보 획득은 매우 어렵다.

기존 성능 평가 방법론의 조인 관련 질의들은 질의 최적기의 전반적인 성능을 알아보는 한 가지 방편으로 사용되거나 특정 조인 알고리즘의 우수성을 밝히기 위한 것이 목적이기 때문에 일반 데이터베이스 시스템 사용자들이 사용하는 조인 질의의 전반적인 수행 능력을 평가하기에는 질의 범위와 질의 평가 항목이 부족하다는 단점을 가진다. 따라서 사용자 관점에서 데이터베이스 시스템의 조인 연산에 대한 성능을 평가할 수 있는 새로운 성능 평가 방법론의 개발이 필요하다.

본 논문은 주어진 사용자 환경에서 조인 연산을 적절하게 수행하는 데이터베이스 시스템을 사용자 관점에서 선택하기 위한 조인 연산 평가 방법론을 기술한다. 본 논문에서는 두 가지 조인 범주(category)와 그에 따른 조인 성능 영향 요소를 고려하여 총42개의 조인 평가 질의를 제안한다. 또한, 본 논문에서는 SIMS(Soongsil Integrated benchMark Suite)[12]를 사용하여 조인 연산 평가 방법론을 두 개의 상용 데이터베이스 시스템을 대상으로 구현하고 평가한다. SIMS는 다양한 데이터베이스 시스템 성능 평가의 용이한 수행을 위해 개발한 통합 성능 평가 도구이다.

본 성능 평가 방법론에서는 조인 연산의 다양성과 상용 데이터베이스 시스템마다 다른 조인 연산 지원 범위, 그리고 성능 평가에 따른 많은 측정 시간 등을 고려하여 두 가지 범주에서의 조인 연산을 측정한다. 두 가지 범주는 다양한 조인 질의 구성을 통한 조인 연산 성능 평가 범주와 대용량 데이터에 기반을 둔 동등 조인 연산 성능 평가 범주이다. 다양한 조인 연산 평가 질의 수행을 통해 여러 종류의 조인 연산을 적절하게 잘 수행하는 데이터베이스 시스템을 사용자가 쉽게 선택할 수 있도록 한다. 또한, 대용량 데이터가 사용되는 현대 업무 환경에서 사용자들이 가장 많이 사용하는 동등 조인 연산을 적합하게 수행할 수 있는 데이터베이스 시스템의 선택을 가능하게 한다.

본 논문의 구성은 다음과 같다. 2장에서는 사용자 관점에서의 조인 연산 평가 방법론을 두 가지 범주로 나누어 각 범주에서 조인 연산 성능에 영향을 미치는 요소, 질의 시나리오, 데이터 구성 및 생성 방법을 기술한다. 3장에서는 SIMS를 통해 구현한 조인 연산 평가 방법론을 두개의 상용 데이터베이스 시스템 상에서 수행하여 조인 연산 성능을 평가하고 결과를 보인다. 마지막으로, 4장에서는 본 논문의 결론을 기술한다.

2. 사용자 관점에서의 조인 연산 평가 방법론

사용자의 조인 연산 성능과 관련된 주 관심 사항은 성능 평가 대상 데이터베이스 시스템에서 다양한 조인 알고리즘 중 어떤 조인 알고리즘을 사용하여 어떻게 수행하는지가 아니라 자신이 수행하고자 하는 조인 연산을 어떤 데이터베이스 시스템이 더 빨리 수행하는가 하는 점이다. 본 평가 방법론에서는 사용자의 관심 대상인 조인 연산 수행 능력에 초점을 맞추어 조인 연산을 대표할 수 있는 측정 항목을 택해 두 가지 범주에서 조인 연산의 성능을 측정한다.

2.1 다양한 조인 질의 구성을 통한 조인 연산 평가

사용자들은 많은 종류의 조인 연산 질의를 수행하기 때문에 조인 연산 평가를 위한 다양한 조인 질의 구성 시 조인 연산 종류를 고려한다. [6]은 동등 조인과 자연 조인(natural join)을 비롯하여 외부 조인(outerjoin), 자체 조인(self join) 등 많은 조인 연산 종류를 정의하고 있다. 하지만, 상용 데이터베이스 시스템은 [6]에서 정의하고 있는 조인 연산 종류를 모두 지원하지는 않기 때문에 대부분의 상용 데이터베이스 시스템이 지원하는 조인 연산을 고려하여 질의를 구성한다. 세타 조인(thetajoin) 중에서 질의 처리 양상이 다른 '<>' 세타 조인, 나머지 세타 조인을 대표하여 '<' 세타 조인, 외부 조인을 대표하여 좌측 외부 조인(left outerjoin)과 사용자들이 가장 많이 사용하는 동등 조인 연산 등 총 4가지 조인 종류를 사용하여 질의를 구성한다.

조인 연산의 대상은 테이블이다. 조인 연산은 대상 테이블의 구성 방법에 따라 성능에 많은 영향을 받는다. 본 방법론에서는 테이블 구성을 테이블에 저장되는 튜플(tuple) 개수에 기반을 두어 수행한다. 조인 테이블의 튜플 수에 따라 조인 연산 수행 시 사용되는 조인 알고리즘이 달라지며, 선택된 조인 알고리즘이 조인 성능에 영향을 미치기 때문이다. 조인 테이블간의 튜플 수 차이가 많이 나는 경우와 조인 테이블이 튜플 수를 많이 가지는 경우로 나누어 조인 테이블을 구성한다. 일반적으로 조인 테이블의 튜플 수가 서로 많이 차이 나는 경우에는 해시 조인 알고리즘을 사용하는 것이 다른 조인 알고리즘을 사용하는 것에 비해 좋은 성능을 보인다[13]. 튜플 수가 많은 테이블 간 조인은 조인 칼럼(column)의 정렬 여부에 따라 해시 조인 알고리즘을 사용하거나 정렬 합병 조인(sort merge join) 알고리즘을 사용하는 것이 중첩 루프 조인(nested loops join) 알고리즘을 사용하는 것보다 효과적이다[14]. 조인의 대상이 되는 테이블은 두 개로 한다.

조인 칼럼에의 인덱스 생성 유무는 조인 알고리즘의 선택 시 주요 고려 대상일 뿐만 아니라 같은 조인 알고리즘을 사용하는 경우에도 큰 성능 차이를 낼 수 있는 영향 요소이다. 특히 중첩 루프 조인 알고리즘을 사용하는 경우 조인 칼럼에 인덱스가 적용되면 인덱스를 사용하지 않은 경우보다 성능

향상 효과를 보인다[2]. 조인 칼럼은 2개며 인덱스는 조인 칼럼에 생성한다. 인덱스 생성 방법은 두 개의 조인 칼럼 모두에 인덱스를 생성하지 않는 경우, 질의 기술 시 'from' 절에 위치한 첫 번째 테이블의 조인 칼럼에만 인덱스를 생성하는 경우, 조인 칼럼 모두에 인덱스를 생성하는 경우로 나뉜다.

본 조인 연산 평가 방법론에서는 앞서 언급한 세 가지 영향 요소 즉, 조인 연산 종류, 조인 대상 테이블의 구성 방법, 조인 칼럼에의 인덱스 생성 여부에 대한 세부 항목을 교차 적용하여 다양한 조인 질의를 구성한다. 질의 수행을 위한 스키마는 200바이트(bytes) 크기의 튜플을 가지는 (그림 1)과 같은 구조의 테이블로 구성된다. (그림 1)에 기술된 각 테이블은 일련 번호 정수 칼럼, 균일 분포 임의 정수 칼럼, 특정 문자열을 가지는 패딩(padding) 칼럼 등 3개의 칼럼을 가진다. 조인 칼럼으로는 일련 번호 정수 칼럼과 균일 분포의 임의 정수 칼럼을 사용한다. 패딩 칼럼을 사용하여 튜플의 크기를 200바이트로 설정한 이유는 지금까지 연구된 대부분의 성능 평가들[8, 9, 10]이 200바이트 또는 그와 유사한 길이로 튜플의 크기를 가지기 때문이다. 'tableC', 'tableD', 'tableE'의 칼럼들은 'tableA'나 'tableB'의 칼럼들과 이름만 다를 뿐 구성 내용은 동일하므로 자세한 기술은 생략한다.

```
create table tableA (
        col1A     integer,     // 일련 번호 정수
        col2A     integer,     // 균일 분포 임의 정수
        chpadA    char(192)    // 패딩 문자
);

create table tableB (
        col1B     integer,     // 일련 번호 정수
        col2B     integer,     // 균일 분포 임의 정수
        chpadB    char(192)    // 패딩 문자
);

create table tableC ( … );
create table tableD ( … );
create table tableE ( … );
```

(그림 1) 다양한 조인 질의 구성을 통한 조인 연산 평가 스키마

시험 데이터베이스 크기는 데이터베이스 시스템의 현재 데이터 버퍼 크기에 비례하여 결정된다. 데이터 버퍼는 데이터베이스 서버에 위치하며, 데이터 버퍼 크기라 함은 데이터베이스 시스템이 데이터 파일에서 읽은 데이터 블록을 저장하기 위해 사용하는 공간의 크기를 의미한다. 데이터 버퍼 크기에 데이터베이스 사용자 프로세스 당 할당되는 개별 영역의 크기는 포함되지 않는다. 데이터 버퍼 크기에 개별 영역 크기가 포함될 경우, 데이터베이스 사용자 프로세스 수에 따라 데이터 버퍼 크기가 유동적으로 변하기 때문이다.

조인 성능 측정 대상 데이터베이스 시스템의 데이터 버퍼 크기가 N MB라 할 때, 'tableC'에 삽입될 텍스트 데이터의 크기가 2N MB가 되도록 'tableC'의 튜플 수를 결정한다. 예를 들어 데이터 버퍼의 크기가 100MB라 하면 텍스트 데이터의 크기는 그 두 배인 200MB여야 하고, 각 튜플의 크기가 200바이트이므로 결국 튜플 수는 100만 개가 된다. 나머지 테이블들의 튜플 수는 'tableC'의 튜플 수에 비례하여 정해진다. 데이터베이스 시스템의 데이터 버퍼 크기가 'N/10000' MB일 때, 각 테이블 별 튜플 개수 및 각 칼럼을 구성하는 값들은 <표 1>과 같다. 'tableD'와 'tableE'은 다른 테이블들과 달리 칼럼 구성 값을 'tableB'와 'tableC'로부터 각각 유도하여 생성한다.

<표 1> 각 테이블 별 튜플 개수 및 칼럼 구성 값

테이블 명	튜플 개수	칼럼 명	칼럼 구성 값
tableA	$\lfloor \frac{N}{1000} \rfloor$	col1A	0 ~ ⌊N/1000⌋-1
		col2A	0 ~ ⌊N/100000⌋-1
		chpadA	192 바이트 임의 문자열
tableB	$\lfloor \frac{N}{10} \rfloor$	col1B	0 ~ ⌊N/10⌋-1
		col2B	0 ~ ⌊N/1000⌋-1
		chpadB	192 바이트 임의 문자열
tableC	N	col1C	0 ~ N-1
		col2C	0 ~ ⌊N/100⌋-1
		chpadC	192 바이트 임의 문자열
tableD	$\lfloor \frac{N}{1000} \rfloor$	col1D	insert into tableD
		col2D	select * from tableB
		chpadD	where col1B between 0 and ⌊N/1000⌋-1
tableE	$\lfloor \frac{N}{100000} \rfloor$	col1E	insert into tableE
		col2E	select * from tableC
		chpadE	where col1C between 0 and ⌊N/100000⌋-1

<표 2> 다양한 조인 질의 구성을 통한 조인 연산 평가 질의

질의 번호	조인 테이블 및 질의 종류	생성 인덱스 수	질의 내용
1	AB equijoin	0	select a.col2A, b.col1B from tableA a, tableB b where a.col1A = b.col2B
5		1	
9		2	
2	AB left outerjoin	0	select a.col2A, c.col2B from tableA a left outer join tableB b on a.col1A = b.col1B
6		1	
10		2	
3	AD '<' thetajoin	0	select a.col2A, d.col1D from tableA a, tableD d where a.col1A < d.col2D
7		1	
11		2	
4	AD '<>' thetajoin	0	select a.col2A, d.col1D from tableA a, tableD d where a.col1A <> d.col2D
8		1	
12		2	
13	BC equijoin	0	select b.col2B, c.col1C from tableB b, tableC c where b.col1B = c.col2C
17		1	
21		2	
14	BC left outerjoin	0	select b.col2B, c.col2C from tableB b left outer join tableC c on b.col1B = c.col1C
18		1	
22		2	
15	BE '<' thetajoin	0	select b.col2B, e.col1E from tableB b, tableE e where b.col1B < e.col2E
19		1	
23		2	
16	BE '<>' thetajoin	0	select b.col2B, e.col1E from tableB b, tableE e where b.col1B <> e.col2E
20		1	
24		2	

다양한 조인 질의 구성을 통한 조인 연산 평가를 위한 질의는 조인 종류 4가지, 튜플 수에 따른 조인 테이블의 구성 방법 2가지, 조인 칼럼의 인덱스 생성 유무 3가지 등 조인 성능에 영향을 미치는 요소의 상세 평가 항목을 교차 적용하여 총24개로 구성된다. 평가 질의는 <표 2>와 같다. <표 2>에서 조인 테이블 및 질의 종류에 기술된 영문 대문자는 <표 1>에 기술된 조인 테이블을 의미하고, 영문 소문자는 조인 종류를 의미한다.

2.2 대용량 데이터에 기반한 동등 조인 연산 평가

데이터 저장 기술의 빠른 발전과 새로운 형태의 데이터 등장으로 인해 사용자들은 대용량 데이터를 자주 접한다. 쉽게 접할 수 있는 대용량 데이터에 기반하여 일반 사용자들이 가장 많이 사용하는 조인 연산 종류인 동등 조인 연산을 평가하는 것은 데이터베이스 시스템 간의 조인 성능 차이가 크게 날 수 있는 환경이라는 점과 사용자가 쉽게 접할 수 있는 환경에서의 조인 연산에 대한 평가라는 점에서 의미가 있다.

대용량 데이터를 생성하기 위해 조인 테이블 당 튜플 수를 고정시키고 테이블의 개수를 증가시키는 방법을 사용한다. 조인 테이블의 개수는 4개부터 9개까지 1.5배씩 증가된다. TPC 성능 평가들에서도 조인 테이블의 개수는 9개를 넘지 않는다. TPC 성능 평가 종류에 따라 다르지만 2개에서 8개까지의 조인 테이블을 대상으로 한 동등 조인 연산을 수행한다. 조인 테이블의 개수가 증가하면 조인 대상 튜플 수가 증가하게 되고, 이로 인해 조인 테이블의 개수가 적을 때보다 조인 성능이 감소하게 된다. 조인 테이블 개수 증가에 따라 조인 성능이 기하급수적으로 감소되는 데이터베이스 시스템보다는 선형적으로 감소되는 데이터베이스 시스템이 보다 좋은 조인 성능을 보이는 시스템이라 할 수 있다.

일반 사용자들은 조인 테이블의 모든 데이터를 대상으로 동등 조인 연산을 수행하는 경우보다 일부 데이터로 한정하여 수행하는 경우가 많다. 조인 범위의 조정은 질의문의 조건절에서 조인 선택률이 기술된 조인 칼럼의 수를 조절함으로 가능하다. 조인 선택률은 해당 테이블의 전체 튜플 수에 대한 조인 참여 튜플 수의 비율이다. 조인 선택률이 기술된 칼럼 개수는 인덱스 생성 유무와 밀접한 연관이 있다. 생성된 인덱스가 조인 칼럼에 적용되는 경우, 인덱스가 적용된 칼럼의 개수가 많아질수록 조인 수행 시간이 감소한다. 조인 선택률이 기술된 조인 칼럼의 개수는 1개, $\lceil N/2 \rceil$개, N개를 사용한다. 'N'은 조인 대상 테이블의 수이다.

인덱스 생성 여부는 앞 절에서 언급된 바와 같이 조인 연산 수행 시 성능 변화에 큰 영향을 미치는 주요한 요소이다. 인덱스 적용을 고려하여 조인 선택률은 10%로 한다. 인덱스를 생성시키고자 하는 칼럼에 각 값들이 균등 분포되었다고 가정할 때, 어떤 값이 테이블에 평균적으로 분포되어 있는 정도가 10%~15%를 넘지 않는 것이 인덱스 적용에 유리하기 때문이다. 인덱스를 생성하는 경우와 인덱스를 생성하지 않는 경우로 나누어 평가 질의를 구성한다. 인덱스를 생성하는 경우에는 조인 선택률이 기술된 조인 칼럼 모두에 대해 수행한다.

동등 조인 연산 평가에서는 앞서 언급한 세 가지 영향 요소 즉, 조인 테이블 개수, 조인 선택률이 기술된 조인 칼럼의 개수, 인덱스 생성 유무의 세부 항목을 교차 적용하여 평가 질의를 구성한다. 질의 수행을 위한 스키마는 200바이트 크기의 튜플을 가지는 테이블 9개로 구성되며, (그림 2)와 같은 구조를 가진다. 스키마를 구성하는 모든 테이블은 일련 번호 정수 칼럼의 위치만 테이블 내에서 변경될 뿐 일련 번호 정수 칼럼과 7개의 20바이트 고정 길이 문자열 패딩 칼럼, 1개의 56바이트 고정 길이 문자열 패딩 칼럼을 가진다. (그림 2)에 기술된 'testtable1'과 'testtable2'의 칼럼 구성을 통해 'testtable3'~'testtable9'의 칼럼 구성이 예상되므로 상세한 기술은 생략한다. 데이터베이스의 크기는 데이터베이스 시스템의 현재 데이터 버퍼 크기에 비례하여 확장 가능하다. 데이터 버퍼 크기의 의미는 앞 절에서 언급한 바와 같다. 각 테이블의 일련 번호 정수 칼럼은 데이터베이스 시스템의 데이터 버퍼 크기가 'N/10000'MB일 때 0에서부터 'N-1'까지의 정수 값을 가진다.

```
create table testtable1 (
        sel1        integer,    // 일련 번호 정수
        chpad20A    char(20),   // 20 바이트 임의 패딩 문자
        chpad20B    char(20),   // 20 바이트 임의 패딩 문자
        chpad20C    char(20),   // 20 바이트 임의 패딩 문자
        chpad20D    char(20),   // 20 바이트 임의 패딩 문자
        chpad20E    char(20),   // 20 바이트 임의 패딩 문자
        chpad20F    char(20),   // 20 바이트 임의 패딩 문자
        chpad20G    char(20),   // 20 바이트 임의 패딩 문자
        chpad56     char(56)    // 56 바이트 임의 패딩 문자
);
create table testtable2 (
        chpad20A    char(20),   // 20 바이트 임의 패딩 문자
        sel2        integer,    // 일련 번호 정수
        chpad20B    char(20),   // 20 바이트 임의 패딩 문자
        …
        chpad56     char(56)    // 56 바이트 임의 패딩 문자
);
create table testtable3 ( … );
…
create table testtable9 ( … );
```

(그림 2) 대용량 데이터에 기반한 동등 조인 연산 평가 스키마

대용량 데이터에 기반한 동등 조인 연산 평가 방법론의 평가 질의는 조인 테이블 개수 3가지, 조인 선택률이 기술된 조인 칼럼 개수 3가지, 인덱스 생성 여부 2가지 등 세 가지 영향 요소의 상세 항목을 교차 적용하여 총18개의 질의로 구성된다. <표 3>은 대용량 데이터에 기반을 둔 동등 조인 연산 평가 질의를 보인다. 평가 질의는 하나의 조인 테이블 수에 대해 6개의 질의로 구성되며, 그 중 3개의 질의는 인덱스를 생성한 경우이며, 나머지 3개 질의는 인덱스를 생성하지 않은 경우이다. 질의 내용 중 조건절에 기술된 'N'은 테이블에 저장된 전체 튜플 수를 의미한다. 평가 질의 중 질의 7~질의 18은 조인 테이블 수와 조인 선택률이 기술된 조인 칼럼 수만 다를 뿐 질의 1~질의 6과 유사하게 구성되므로 공간 제약 상 생략한다.

〈표 3〉 대용량 데이터에 기반을 둔 동등 조인 연산 평가 질의

질의 번호	조인 테이블 수	조인 선택률이 기술된 조인 칼럼 수	인덱스 생성 여부	질의 내용
1	4	1	비 생성	select sel1, sel2, sel3, sel4 from testtable1, testtable2, testtable3, testtable4 where sel1 = sel2 and sel2 = sel3 and sel3 = sel4 and sel1 between 0 and (N / 10-1)
2		2		select sel1, sel2, sel3, sel4 from testtable1, testtable2, testtable3, testtable4 where sel1 = sel2 and sel2 = sel3 and sel3 = sel4 and sel1 between 0 and (N / 10-1) and sel2 between 0 and (N / 10-1)
3		4		select sel1, sel2, sel3, sel4 from testtable1, testtable2, testtable3, testtable4 where sel1 = sel2 and sel2 = sel3 and sel3 = sel4 and sel1 between 0 and (N / 10-1) and sel2 between 0 and (N / 10-1) and sel3 between 0 and (N / 10-1) and sel4 between 0 and (N / 10-1)
4		1	생성	질의 1과 질의 내용 동일
5		2		질의 2와 질의 내용 동일
6		4		질의 3과 질의 내용 동일

2.3 조인 연산 성능 측정 방법

성능 평가 시험은 일반적인 사용 목적(general purpose)의 표준 상태로 설치된 데이터베이스 시스템을 대상으로 한다. 일반적인 사용 목적의 표준 설치는 사용자들이 일반적인 업무 환경에서 손쉽고 효과적으로 해당 데이터베이스 시스템을 사용하기 위해 수행한다. 데이터베이스 시스템의 많은 파라미터들 중 조인 연산과 연관된 파라미터들을 시험의 공정성을 위해 비교 대상 데이터베이스 시스템 모두에 동등하게 설정하는 작업이 매우 어렵기 때문이다.

데이터베이스 시스템마다 설정 가능한 파라미터의 종류가 다르고, 파라미터 기본 값 또한 다르기 때문에 표준 설치 상태에서는 비교 대상 데이터베이스 시스템간에 기본적인 성능 차이가 발생할 수 있다. 본 성능 평가 방법론에서는 데이터베이스 시스템간의 공정한 성능 평가 시험을 위해 데이터베이스 크기를 사용한 핸디캡(handicap)을 적용한다. 시험 질의의 반복 수행을 통해 평균 질의 응답 시간을 계산하는 성능 측정 방법에 가장 큰 영향을 미치는 데이터 버퍼 크기에 비례하여 데이터베이스 크기를 조정하는 방법을 사용한다. 즉, 데이터 버퍼 크기가 크게 설정된 데이터베이스 시스템은 그만큼 큰 데이터베이스를 대상으로 시험 질의를 수행하고, 데이터 버퍼 크기가 작게 설정된 시스템은 작은 데이터베이스를 대상으로 시험 질의를 수행한다.

시험 방법은 각 질의 당 5회를 수행하여 초기 2회 질의 수행 시간을 제외한 나머지 3회 질의 수행 시간의 산술 평균값을 측정값으로 가지는 웜 스타트(warm start) 방법을 사용한다. <표 2>와 <표 3>에 기술된 각 질의들의 수행 시간은 1/1000초 단위로 측정되어 양의 소수형(decimal type) 실수로 표현한다.

3. 조인 연산 평가 시험 및 결과

본 논문에서는 사용자 관점에서의 조인 연산 평가 방법론을 두 개의 상용 데이터베이스 시스템 상에서 SIMS를 이용하여 구현하였다. 시험에 사용된 데이터베이스 시스템의 이름은 라이선스(license) 문제로 인해 명기하기 곤란하므로 본 논문에서는 각각의 데이터베이스 시스템을 시스템 X와 시스템 Y로 명명한다. 시험은 1.6GHz 펜티엄4 프로세서가 장착된 PC에서 수행되었다. 시험 수행에 사용된 PC는 1GB 메모리와 총190GB 하드 디스크 공간을 가지고 있고, 운영 체제로는 와우 리눅스7.3 파란 릴리즈2를 사용하고 있다.

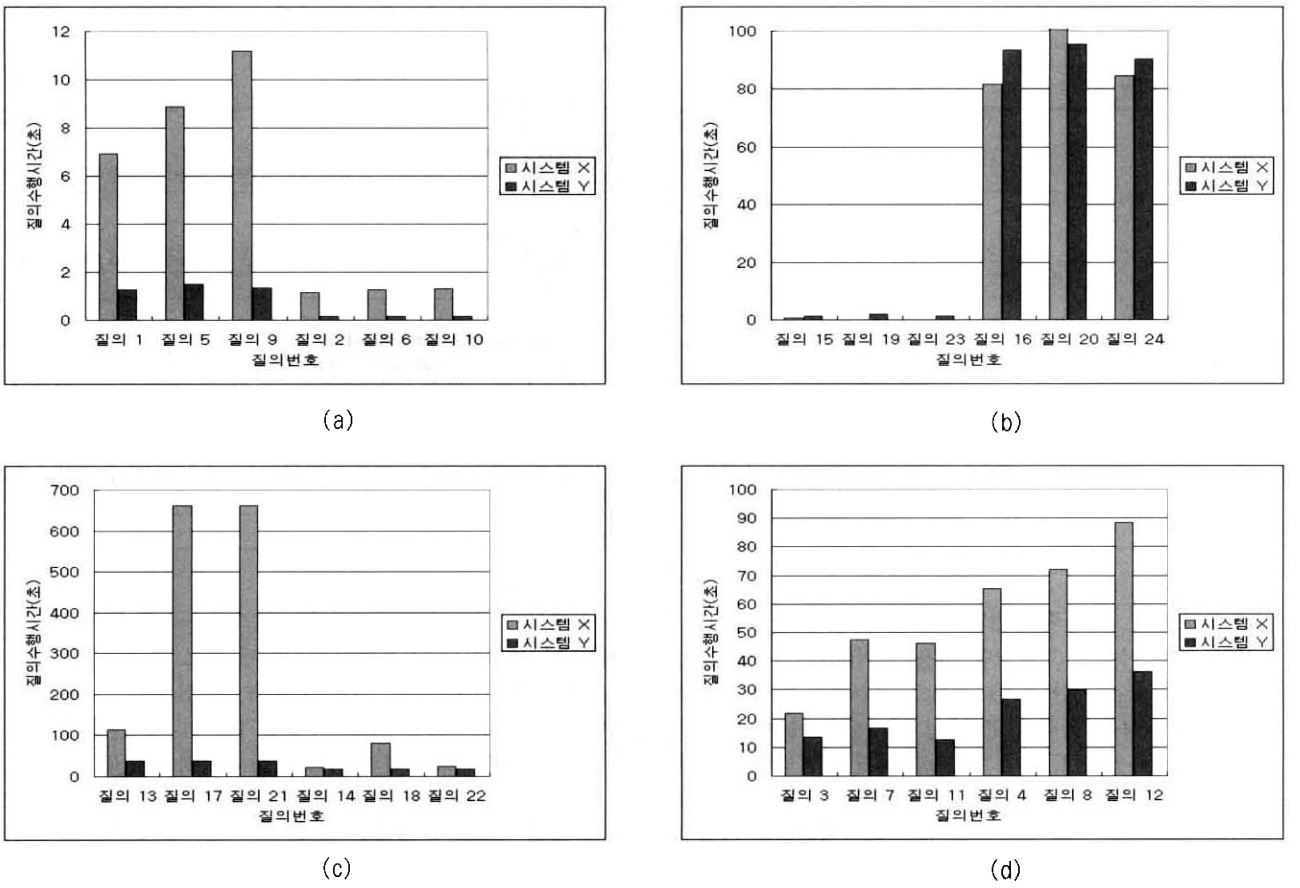
시험 대상 데이터베이스 시스템은 일반 사용 목적으로 설치되었고, 설치 시 설정된 파라미터 값은 수정되지 않았다. 시험 데이터베이스 크기를 결정하는 각 데이터베이스 시스템의 데이터 버퍼 크기는 시스템 X가 80MB, 시스템 Y가 208MB이다. 데이터를 저장하기 위한 영역으로 시스템 X는 20GB, 시스템 Y는 40GB를 가진다. 인덱스를 저장하기 위한 영역과 임시 공간으로 사용하기 위한 영역은 두 시스템 모두에 각각 2GB씩 할당되었다.

각 시험 범주 별로 테이블들이 가지는 튜플 수는 각 데이터베이스 시스템에 설정된 데이터 버퍼 크기에 비례하여 <표 4>와 같이 설정하였다.

〈표 4〉 조인 연산 평가 시험에 사용된 각 테이블 당 튜플 수

시험 범주	테이블 종류	시스템 X(개)	시스템 Y(개)
다양한 조인 질의 구성을 통한 조인 연산 평가	tableA	800	2,080
	tableB	80,000	208,000
	tableC	800,000	2,080,000
	tableD	800	2,080
	tableE	8	20
대용량 데이터에 기반을 둔 동등 조인 연산 평가	모든 테이블	800,000	2,080,000

다양한 조인 질의 구성을 통한 조인 성능 비교 시험의 결과는 (그림 3)과 같다. 시스템 X의 질의 20에 대한 수행 시간은 521.228초로 측정되었으나 다른 질의들과의 시간차가 커서 (b)의 막대그래프에 전체 값을 표기하지 못하고 100초가 넘는 것으로만 표기하였다. 또한, (b)에서 질의 19와 질의 23에 대한 시스템 X의 수행 시간은 다른 질의 수행 시간에 비해 매우 작게 나타나 막대그래프에 거의 표현되지 않았다.



(그림 3) 다양한 조인 질의 구성을 통한 조인 성능 평가 시험 결과

■ 튜플 수 차이가 큰 테이블 간에 수행된 동등 조인 연산과 좌측 외부 조인 연산 평가

(그림 3)의 (a)는 튜플 수 차이가 100배인 테이블 간에 수행된 동등 조인 연산과 좌측 외부 조인 연산 평가 결과를 보인다. 동등 조인 연산 수행 결과를 보이는 질의 1, 질의 5, 질의 9에서는 시스템 X에 비해 시스템 Y가 약5.4배에서 8.4배 더 나은 성능을 보인다. 또한, 튜플 수 차이가 큰 테이블 간의 좌측 외부 조인을 나타내는 질의 2, 질의 6, 질의 10에서도 시스템 Y가 시스템 X에 비해 약7.1배에서 7.9배까지 더 나은 조인 성능을 보인다.

■ 튜플 수 차이가 큰 테이블 간에 수행된 세타 조인 성능 비교

튜플 수 차이가 1000배인 테이블 간에 수행된 세타 조인 연산 평가 결과는 (그림 3)의 (b)에 보인다. '<>' 세타 조인을 수행하는 질의 20을 제외한 나머지 경우에는 시스템 X가 시스템 Y에 비해 좋은 조인 성능을 보였으며, 적게는 1.1배에서 많게는 221.0배까지 차이를 나타냈다. '<>' 세타 조인에서는 시스템 Y가 시스템 X에 비해 5.5배 더 나은 조인 성능을 보였다. 시스템 X는 질의 19, 질의 23을 수행할 때 생성된 인덱스를 활용하였으며, 인덱스를 활용한 경우가 인덱스를 사용하지 않은 질의 15의 경우에 비해 약33.8배~129.5배까지 좋은 성능을 보였다.

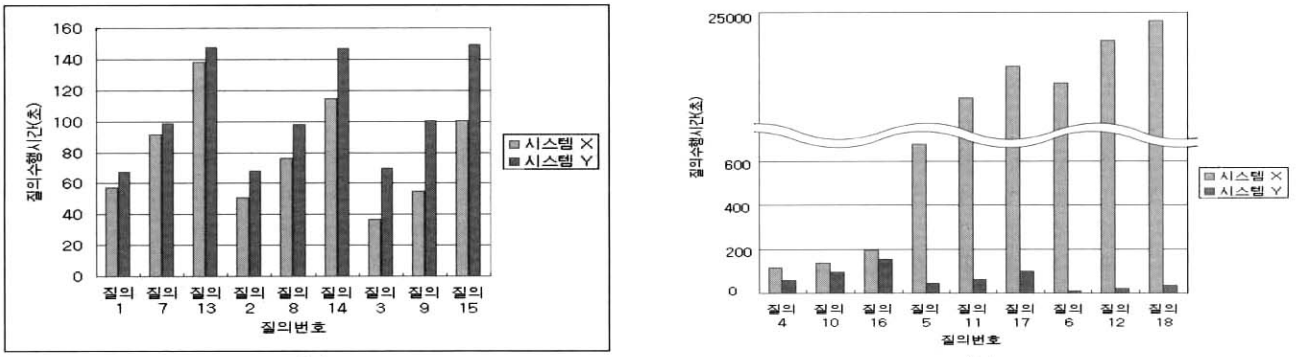
■ 튜플 수가 많은 테이블 간에 수행된 동등 조인 연산 평가

(그림 3)의 (c)는 튜플 수가 많은 테이블 간 동등 조인 연산과 좌측 외부 조인 연산에 대한 평가 결과를 나타낸다. 동등 조인 질의나 좌측 외부 조인 질의 모두 시스템 Y가 시스템 X에 비해 약1.2배에서 17.5배까지 좋은 조인 성능을 나타냈다.

■ 튜플 수가 많은 테이블 간에 수행된 세타 조인 연산 평가

(그림 3)의 (d)는 튜플 수가 많은 테이블에 대해 수행된 세타 조인 연산 평가 결과를 보인다. 인덱스 활용 양상이 두 시스템 모두 비슷하나, 조인 성능에서는 (b)에서 보인 결과와 정반대로 시스템 Y가 시스템 X에 비해 약1.6배에서 3.6배 더 나은 조인 성능을 보인다.

(그림 3)에서 살펴본 바와 같이 시스템 Y가 시스템 X에 비해 전반적으로 조인 연산 수행 능력이 우수하다. 튜플 수 차이가 큰 테이블간에 세타 조인을 수행하는 경우에는 '<>' 세타 조인을 제외하고 시스템 X가 시스템 Y보다 더 나은 세타 조인 연산 수행 능력을 보임을 알 수 있다. 조인 연산 평가 방법론은 다양한 조인 질의를 구성하여 수행함에 따라 전반적인 조인 연산 능력은 물론 특정 상황에서 특정 데이터베이스 시스템이 더 나은 성능을 발휘할 수 있음을 알아낼 수 있기 때문에 사용자가 특정 형태의 조인 질의를 많이 사용하는 환경에서 더 나은 조인 연산 성능을 보이는 데이터베이스 시스템을 선택하는데 도움이 된다.



(그림 4) 대용량 데이터에 기반한 동등 조인 성능 비교 시험 결과

대용량 데이터에 기반한 동등 조인 성능 비교 시험 결과는 (그림 4)와 같다.

■ 인덱스를 생성하지 않은 경우 대용량 데이터에 기반한 동등 조인 연산 평가

(그림 4)의 (a)에서 질의 1, 질의 7, 질의 13과 질의 2, 질의 8, 질의 14, 그리고 질의 3, 질의 9, 질의 15를 그룹 지어 살펴보면 각 그룹 내에서 질의 번호가 증가할수록 더 많은 조인 테이블을 대상으로 한 조인 질의다. 즉, 질의 1은 4개 조인 테이블을 대상으로 조인을 수행하고, 질의 7은 6개, 질의 13은 9개의 조인 테이블을 대상으로 동등 조인을 수행한 결과이다. 각 그룹 내에서 질의 테이블의 개수가 증가할수록 시스템 X와 시스템 Y 모두 질의 수행 시간이 증가됨을 알 수 있다. 두 시스템간의 질의 수행 시간 증가 비율은 큰 차이를 보이지는 않았다.

인덱스를 생성하지 않은 경우 시스템 X와 시스템 Y의 동등 조인 연산 성능 차이는 조인 선택률이 기술된 조인 칼럼의 개수에 따라서 발생하였다. 조인 선택률이 기술된 조인 칼럼이 1개 사용된 질의 1, 질의 7, 질의 13에서보다 조인 선택률이 기술된 조인 칼럼이 $\lceil N/2 \rceil$개 사용된 질의 2, 질의 8, 질의 14에서의 두 시스템 간 성능 차가 더 크게 발생하였고, 조인 선택률이 기술된 조인 칼럼이 N개 사용된 질의 3, 질의 9, 질의 15에서는 이전 결과보다 더욱 심한 조인 성능 차가 발생하였다. 'N'은 조인 테이블의 개수를 의미한다. 조인 선택률이 기술된 조인 칼럼의 개수가 증가할수록 시스템 X의 질의 수행 시간은 감소하나, 시스템 Y의 질의 수행 시간에는 큰 변화가 없기 때문이다. 인덱스를 생성하지 않은 경우 시스템 X는 시스템 Y에 비해 약1.1배에서 약 1.9배 더 나은 동등 조인 성능을 보인다.

■ 인덱스 생성 시 대용량 데이터에 기반한 동등 조인 연산 평가

(그림 4)의 (b)의 결과를 보면 (a)에서의 결과와 마찬가지로 조인 테이블이 증가할수록 두 시스템 모두 질의 수행 시간이 증가함을 알 수 있다. 그러나 인덱스가 생성된 경우에는 인덱스가 생성되지 않은 경우와 상반된 결과를 보인다. 조건절에서의 조인 선택률이 기술된 조인 칼럼의 개수가 증가할수록 시스템 Y가 시스템 X에 비해 더 좋은 성능을 보인다. 이는 시스템 X가 조인 선택률이 기술된 조인 칼럼의 개수가 증가할수록 질의 수행 시간이 증가하는데 반해 시스템 Y는 인덱스의 효율적인 사용을 통해 수행 시간의 점진적인 감소를 보이기 때문이다. 인덱스를 생성한 경우 시스템 Y는 시스템 X에 비해 약 1.2배에서 약811.6배 더 나은 동등 조인 성능을 보인다.

인덱스 생성 유무에 따라 두 데이터베이스 시스템의 대용량 데이터에 기반한 동등 조인 연산 성능 우열이 다르게 나타남을 알 수 있다. 또한, 시스템 X가 인덱스가 생성되지 않은 경우 조인 선택률이 기술된 조인 칼럼의 개수에 따라 질의 수행 시간이 감소하지만, 인덱스가 생성된 경우에는 그와 반대로 질의 수행 시간이 증가함을 알 수 있어 시험 환경과 유사한 작업 환경에서 대부분의 조인 연산을 수행해야 하는 사용자들은 경우에 따라 적절한 데이터베이스 시스템을 선택할 수 있도록 본 조인 연산 평가 방법론이 도움을 준다.

4. 결 론

본 논문에서는 데이터베이스 시스템의 조인 연산에 대한 사용자 관점에서의 평가 방법론을 기술하였다. 기존 성능 평가 방법론의 조인 질의들은 개발자 관점에서 접근하여 우수한 조인 알고리즘을 규명하는데 초점을 맞추었지만, 본 방법론에서는 조인 연산을 빈번히 수행하는 사용자가 자신의 업무에 적합한 조인 연산을 수행하는 데이터베이스 시스템을 선택할 수 있도록 하는데 초점을 맞추었다. 이를 위해 다양한 조인 질의 구성을 통한 조인 연산 성능 평가 범주와 대용량 데이터에 기반을 둔 동등 조인 성능 평가의 범주에서 각각 성능 평가 항목을 선정하여 총42개의 평가 질의를 제안하였다. 또한 본 성능 평가 방법론을 두 개의 상용 데이터베이스 시스템을 대상으로 구현하였고, 성능 평가를 수행하여 그 결과를 기술하였다.

앞서 언급한 바와 같이 지금까지의 많은 관련 연구들은 그 주된 내용이 조인 알고리즘의 개발이나 개선과 관련된 내용이다. 이는 조인 연산에 있어 조인 알고리즘의 중요성을 보여준다. 조인 연산 성능 평가 시험을 통해서도 조인 알고리즘이 성능에 영향을 미치는 중요한 요소임을 알 수 있었다. 다양한 조인 질의 구성을 통한 조인 연산 평가 범주에서 튜플 수 차이가 큰 테이블간의 동등 조인 연산 수행

시 해시 조인 알고리즘을 사용한 시스템 Y가 정렬 합병 조인 알고리즘이나 중첩 루프 조인 알고리즘을 사용한 시스템 X보다 더 나은 성능을 보였다. 튜플 수가 많은 테이블간에 수행된 세타 조인 연산 평가에서는 중첩 루프 조인 알고리즘을 사용한 시스템 X보다 정렬 합병 조인 알고리즘을 사용한 시스템 Y가 더 나은 성능을 보였다.

사용자들이 조인 연산 성능 평가 시험을 수행한 후 시험 결과를 좀 더 쉽게 이해하고 시험 대상 데이터베이스 시스템 간의 전반적인 조인 성능 우위를 손쉽게 판단하기 위해서는 단일 성능 평가 측정값의 도출이 필요하다. 사용자가 자신들의 업무 영역에 적합한 조인 성능을 보이는 데이터베이스 시스템을 선택할 수 있도록 각 질의 별 가중치를 정하고, 측정된 질의 수행 시간 값에 가중치를 곱한 후 모든 가중 질의 수행 시간 값의 합을 구하여 단일 성능 평가 측정값을 계산할 수 있다.

조인 연산 성능에 영향을 미치는 요소로 전체 테이블 읽기(full table scan) 기능을 들 수 있다. 인덱스를 활용할 수 없는 상황에서는 전체 테이블 읽기 기능을 사용하여 조인 연산이 수행되고, 조인 연산 수행 시간 중 많은 부분이 전체 테이블 읽기 작업에 소요되기 때문이다. 향후 전체 테이블 읽기 기능을 포함하는 조인 연산을 구체적으로 정의하여 전체 테이블 읽기 기능이 조인 연산 수행에 미치는 영향 정도를 사용자들이 쉽게 파악할 수 있도록 하고, 이를 통해 자신의 조인 환경에서 적절하게 수행될 수 있는 데이터베이스 시스템을 선택하는데 더욱 도움이 되도록 한다. 또한, XML 데이터베이스 시스템에서의 조인 연산에도 적용 가능하도록 조인 연산 평가 방법론의 확장이 필요하다.

참 고 문 헌

[1] Y. Kambayashi, "Processing Cyclic Queries, In : Query Processing in Database Systems," W. Kim, D. S. Reiner, and D.S. Batory Ed., Springer-Verlag, pp.62-78, 1985.

[2] M. W. Balsgen and K. P. Eswaran, "Storage and Access in Relational Databases," IBM Systems Journal, Vol.16, No.4, pp.363-377, 1977.

[3] T. Nakayama, M. Hirakawa, and T. Ichikawa, "Architecture and Algorithm for Parallel Execution of a Join Operation," Proceedings of the 1st International Conference on Data engineering, pp.160-166, 1984.

[4] L. F. Mackert and G. M. Lohman, "R* Optimizer : Validation and Performance Evaluation for Distributed Queries," Proceedings of the 12th International Conference on Very Large Data Bases, pp.149-159, 1986.

[5] J. M. Patel and D. J. DeWitt, "Partition Based Spatial-Merge Join," Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.259-270, 1996.

[6] R. Mishra and M. H. Eich, "Join Processing in Relational Databases," ACM Computing Survey, Vol.24, No.1, pp.63-113, 1992.

[7] Y. Yang and M. Singhal, "A Comprehensive Survey of Join Techniques in Relational Databases," Computer and Information Science TR. 48, The Ohio State University, 1997.

[8] D. J. DeWitt, "The Wisconsin Benchmark : Past, Present, and Future," In : The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., J. Gray Ed., pp.269-316, Morgan Kaufmann, 1993.

[9] P. O'Neil, "The Set Query Benchmark," In : The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., J. Gray Ed., pp.359-396, Morgan Kaufmann, 1993.

[10] S. H. Lee, S. J. Kim, and W. Kim, "The BORD Benchmark for Object-Relational Databases," Proceedings of the 11th Database and Expert Systems Applications Conference, pp.6-20, 2000.

[11] The TPC home page, http://www.tpc.org/.

[12] H. J. Jeong and S. H. Lee, "An Integrated Benchmark Suite for Database Systems," Proceedings of the IASTED International Conference on Information Systems and Databases, pp.74-79, 2002.

[13] G. Graefe, A. Linville, and L. D. Shapiro, "Sort versus Hash Revisited," IEEE Transactions on Knowledge and Data Engineering, Vol.6, No.6, pp.934-944, 1994.

[14] Microsoft Korea Co., 조인 기술의 비교, http://support.microsoft.com/korea/, 1999.

정 회 진

e-mail : sinclear@dreamwiz.com
1993년 우석대학교 전산학과(학사)
1995년 숭실대학교 대학원 컴퓨터학과(석사)
1995년~2000년 (주)핸디소프트 기술연구소, 선임연구원
2002년~2003년 숭실대학교 정보미디어 연구소, 전임연구원
2000년~현재 숭실대학교 대학원 컴퓨터학과 박사과정
관심분야 : 데이터베이스 시스템 성능 평가 및 튜닝, XML 데이터베이스

이 상 호

e-mail : shlee@comp.ssu.ac.kr
1984년 서울대학교 전산공학과(학사)
1986년 미국 노스웨스턴대 전산학과(석사)
1989년 미국 노스웨스턴대 전산학과(박사)
1990년~1992년 한국전자통신연구원
1999년~2000년 미국 조지메이슨대, 소프트웨어정보공학과, 교환 교수
1992년~현재 숭실대학교 컴퓨터학부 교수
관심분야 : 인터넷 데이타베이스, 데이타베이스 시스템 성능 평가 및 튜닝