# 전역 스키마 생성 도구를 이용한 데이터베이스 통합 및 질의 시스템☆

# Design of Database Integration System and Query System based on Global View Generation Tool

박 우 창*
U-chang Park

## 요 약

기업에서 데이터베이스, 데이터웨어하우스, 데이터마트, OLAP 등의 활용이 증가함에 따라 데이터베이스 통합은 일반적이며 해결해야하는 문제가 되었다. 데이터베이스를 통합하여 공유하는 방법은 여러 가지가 있지만 데이터베이스 시스템의 진정한 상호작동은 각 시스템의 데이터베이스를 의미적으로 비교하여 합병하여야 한다.

본 연구는 기존의 데이터베이스 연합 방식을 개선하여 데이터베이스 관리자가 데이터베이스의 의미를 쉽고 효율적으로 찾아내는 통합 시스템을 제안한다. 연구에서 데이터베이스 의미 정보 통합은 전역 뷰를 생성하는 도구를 사용하여 해결하였다. 전역 뷰를 구성하는 것은 복잡한 지식과 경험을 요구하기 때문에 아직까지 구성 도구가 개발된 사례가 많지 않은 핵심적인 문제이다. 제안된 기법과 도구는 데이터베이스 관리자가 사용하기에 간단하고 고기능이며 대부분의 통합 질의를 지원하도록 충분한 표현력을 갖도록 하였다.

## Abstract

Database integration is a common and growing challenge with the proliferation of database systems, data warehouses, data marts, and other OLAP systems in organizations. Although there are many methods of sharing data between databases, true interoperability of database systems requires capturing, comparing, and merging the semantics of each system. In this work, we present a database integration system that improves on the database federation architecture by allowing domain administrators to simply and efficiently capture database semantics. The semantic information is combined using a tool for producing a global view. Building the global view is the bottleneck in integration because there are few tools that support its construction, and these tools often require sophisticated knowledge and experience to operate properly. The technique and tool presented is simple and powerful enough to be used by all database administrators, yet expressive enough to support the majority of integration queries.

Keyword : database, integration, mediator, view

## 1. 서 론

비즈니스에 필요한 정보를 데이터베이스로 구축하면서 CRM(Customer Relationship Management), ERP(Enterprise Resource Planning), SCM(Supply Chain Management) 등 다양한 데이터베이스 어플리케이션이 도입되어 기업 정보화를 추진해 왔다. 기업들은 비즈니스 유지에 필요한 모든 정보를 비롯하여 고객이 찾고자 하는 모든 정보를 하나의 통합된 뷰를 통해서 제공할 수 있는 인프라 구축이 시급하다. 효과적인 정보자산 관리를 통해 기업의 경쟁력을 높이기 위해서는 기존의 어플리케이션을 유지하면서 빠르고 확장성이 뛰어난 데이터 통합 솔루션이 필요하며, 이러한 통합이 기

* 정 회 원 : 덕성여자대학교 정보공학대학 교수
ucpark@duksung.ac.kr
[2006/12/24 투고 - 2007/01/29 심사 - 2007/03/21 심사완료]
☆ 본 연구는 덕성여자대학교 2007년도 교내연구비 지원에 의해 수행되었음

업의 업무 시간과 비용을 줄이는데 필수적이다. 그러나 독립적으로 구축된 데이터베이스를 통합하여 정보를 추출하기에는 많은 구조적인 제약과 관리의 어려움이 존재한다[1].

이질적이고 자치성을 갖는 데이터베이스 시스템의 통합에 대한 연구는 1980년부터 시작하였다. 그러나 연합 데이터베이스[2]와 분산 데이터베이스를 포함한 초기의 접근 방법은 통합 문제를 완벽히 해결하지는 못하였다. 분산 데이터베이스는 데이터베이스 시스템이 서로 다른 기관에 소속되어 있을 때 지역 자치성을 보장하지 못하였다. 연합 데이터베이스는 데이터에 대한 분산 접근을 시도하지만 질의 형성과정에 통합이 수행될 것을 필요로 한다. 이러한 방법보다 더 실제적인 사용되었던 해결책들은 수동적인 통합, 웹 기반 사용자 인터페이스, 응용 수준 통합(작은 규모의 통합에 적합), SQL 미들웨어를 통한 통합, 지역 통합을 통한 뷰 제공, 공유 저장소의 방법 등이다. 통합 기법의 최근 추세는 데이터 웨어하우스, 데이터 마이그레이션, 응용 프로그램 수준의 통합 등이 있다. 그러나 이러한 기법들은 지역자치성이나 혹은 실시간 데이터베이스 접근을 허용하지 않는다.

데이터베이스 통합에 관한 현재의 기술들은 기본적으로 연합모델[1]에 기초하고 있다. 데이터베이스 연합모델은 메디에이터(mediator)와 표준 프로토콜을 이용하여 시스템간의 데이터를 공유한다. 이 시스템들은 공유를 통하여 분산 질의를 처리하지만 지역 질의 단위로 통합을 수행한다. 또 통합과 지역 데이터베이스 질의 작성이 어렵고 관리가 어려우며 기존 데이터베이스 엔진에서 수행이 되어야한다. 메디에이터와 같이 사용되는 래퍼(wrapper) 구조는 데이터베이스 자치성을 유지하며 분산 질의를 지원한다. 본 연구에서 제안하는 더 효율적인 방법은 데이터 소스로부터 먼저 전역 뷰를 구성하는 것이다. 전역 뷰 구축은 스키마 매칭과 머지 기법을 필요로 한다[3]. 많은 프로토타입 시스템이 구축된 바 있지만 기존 구축 도구들은 데이터베이스 관리자나 설계자가 전역 뷰를 생성하거나 사용하기가 쉽지 않다. 또 아직은 메디에이터와 래퍼 컴포넌트에 대한 표준 인프라구조, 프로토콜의 구현이 없다. 따라서 직접 이러한 컴포넌트를 작성해야하므로 상당한 노력을 필요로 한다[4].

본 논문은 데이터베이스 관리자가 전역 뷰를 구축하는 것을 지원하는 데이터베이스 통합시스템을 제안한다. 시스템을 통하여 구축된 전역 뷰는 데이터베이스 연합시 복잡한 통합 질의 작성 노력을 줄여주고 기존의 접근 방법 보다는 쉽게 전역 뷰를 구성할 수 있다. 전역 뷰 구축 후 데이터 소스에서 오는 질의를 처리하기 위하여 UnityJDBC[5, 6]를 사용하였다. 지역 데이터베이스에 보내진 질의 처리 결과는 단일 결과를 생성하기 위하여 전역 질의 인터페이스에 모아진다. 시스템은 여러 개의 데이터 소스에서 결과를 가져올 수 있으며 JDBC를 이용하여 지역 데이터베이스의 자치성과 투명성을 유지하면서 결과를 통합한다.

기존의 데이터베이스 연합 방식과 비교한 본 연구의 차별점은

- SQL에 관한 기본 지식만으로도 전역 뷰를 구축할 수 있도록 하였고,
- 지역 데이터베이스에 대한 지식이 없이 전역 뷰만으로 사용자가 질의를 할 수 있는 질의 인터페이스를 제공하였으며
- 전역 뷰에서 보낸 질의에 따라 지역 데이터베이스 질의 처리 결과를 자동적으로 결합하는 질의 처리 시스템을 제공하였다.

본 논문의 구성은 다음과 같다. 2장에서 시스템의 전체 구조를 살펴보고, 3장에서는 여러 데이터 소스 스키마에서 전역 뷰를 구성하는 기본적인 구조에 대하여 논하고 전역 뷰를 이용한 질의가 어떻게 지역 질의로 바뀌어 계산되고 단일 결과로 통합되는지 논한다. 4장에서는 제안된 시스

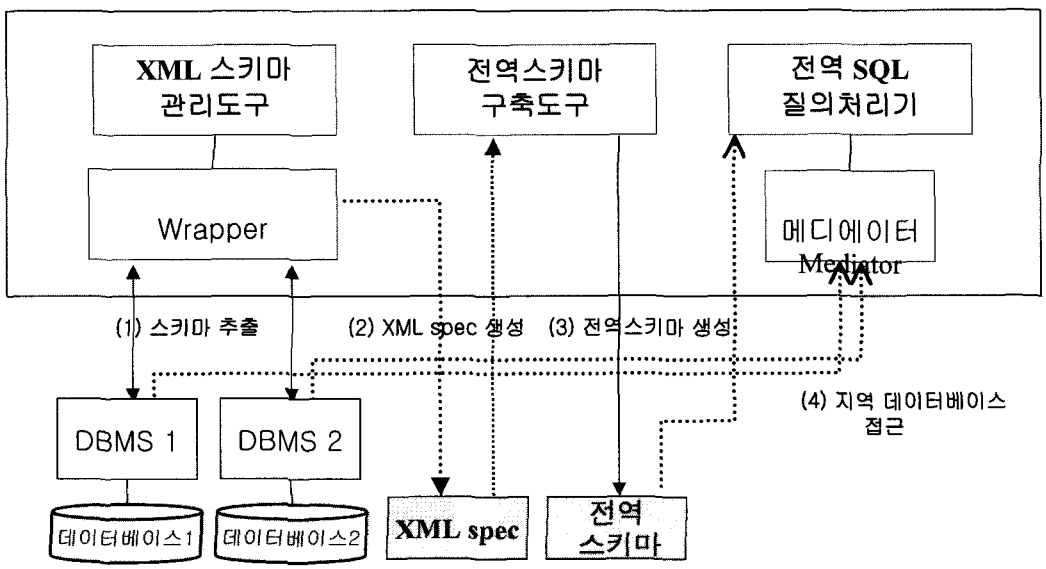
템과 타 시스템의 기능 비교를 하며 마지막 장에서는 향후 연구 방향과 결과를 논한다.

## 2. 시스템 구조

전역 뷰 구축과 질의 시스템에서의 각 구성요소간의 연결 구조는 그림 1과 같다.

시스템 구성도는 지역 데이터베이스에 대한 정보를 수집하는 스키마 추출기(XML Schema Manager), 전역 뷰 생성에 필요한 전역 뷰 편집기(Global Schema Editor), 지역 데이터베이스에 대한 질의 통합을 수행하는 질의처리 인터페이스(*SQL Query Interface*)에 관한 전체 시스템 구조를 보인다. 각 지역 데이터 소스에 대한 질의와 결과 추출은 UnityJDBC[6]로 수행된다. 본 연구의 초점은 전역 뷰의 생성과 전역 질의를 연합 질의로 매핑시키는 것이다. 연합 질의 수행은 다른 시스템을 이용하여 실행이 가능하지만 본 연구에서는 UnityJDBC를 사용하였다. UnityJDBC는 완전 자바 기반의 분산 질의 처리구조이다.

전역 뷰를 구성하기 전 먼저 시스템은 각 지역 데이터베이스 스키마 정보를 모아서 XML 파일로 변환한다(1단계, 2단계). 이질성이 있는 각 데이터베이스 시스템의 스키마 정보를 표현하는 공통된 양식으로 XML을 사용하였다. 다음 단계는 전역 뷰 편집기를 이용하여 전역 뷰를 생성한다. 전역 뷰는 상거래, 문서 정보 혹은 주문 트랜잭션과 같이 이미 어느 정도 표준화되어 존재할 수도 있다. 전역 뷰와 매핑 정보는 일단 구축이 되면 XML 스키마 매핑 파일에 저장된다(3단계). 지역과 전역스키마와의 매핑은 엔티티 단위로 이루어진다. 즉 지역 뷰의 각 테이블과 속성은 전역 뷰의 테이블 혹은 속성과 매핑된다. 3.2절에서 매핑 규칙과 구조를 다룬다. 질의 처리 단계에서 사용자는 GUI를 통하여 전역 뷰에 대한 전역질의를 구성하며, 전역 질의는 질의처리기에 의하여 지역 질의로 변환된다. 지역질의는 연합질의 처리 엔진(UnityJDBC)을 통하여 각 지역 데이터베이스 시스템에 전달되며 처리 후 단일 결과로 결합된다(단계 4, 단계 5). SQL 질의는 XML 스키마 매핑 파일을 이용하여 전역 뷰 정보를 참조하며 실시간으로 지역 데이터베이스의 정보를 가져온다.

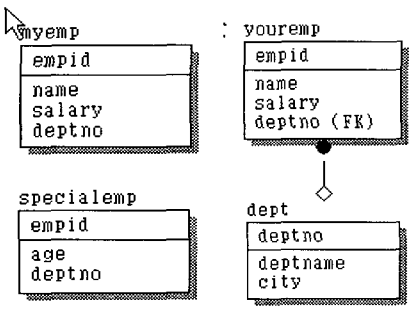



〔그림 2〕 데이터베이스 통합 시스템 구조

# 3. 통합 시스템의 설계

## 3.1 데이터 의미 정보 추출

전역 뷰 생성 전에 지역 데이터베이스의 스키마 정보와 지역 데이터의 추가적인 의미 정보를 추출할 필요가 있다. 제안된 시스템은 각 지역 데이터베이스에 접속하여 스키마 정보를 XML 파일로 저장한다. 각 지역 데이터베이스에 대한 XML 파일은 테이블과 속성의 이름을 XML 원소의 계층구조에 저장한다. 예를 들면 그림 2는 Employee 데이터베이스에 대한 2개의 지역 데이터베이스 스키마를(각각 사이트 1과 사이트 2로 부른다),(.) 그림 3은 사이트 1의 Employee 데이터베이스 스키마의 앞부분을 XML 파일로 저장한 모습을 보인 것이다.




(사이트 1) (사이트 2)

〔그림 2〕 Employee 데이터베이스 스키마 예

```
<TABLE>
    <RefName>site1.myemp</RefName>
    <tableName>myemp</tableName>
    <schemaName></schemaName>
    <catalogName>site1</catalogName>
    <comment>MySQL table</comment>
    <FIELD>
        <RefName>myemp.deptno</RefName>
            <fieldName>deptno</fieldName>
            <dataType>4</dataType>
            <dataTypeName>integer
                        unsigned</dataTypeName>
            <fieldSize>10</fieldSize>
            <decimalDigits>0</decimalDigits>
          ... (생략)
```

〔그림 3〕 데이터베이스 스키마 파일 예 - 사이트 1

## 3.2 전역 뷰 생성

전역 뷰 편집기는 그림 4와 같다. 전역 뷰는 지역 데이터베이스의 개체들을 전역 뷰에 매핑을 시켜 구축한다. 지역 스키마가 독립적으로 구축되기 때문에 전역 뷰와 완벽하게 매치되지 않을 수도 있다. 이 경우 관리자가 새로운 전역 관계를 생성하거나 기존의 전역스키마를 편집할 수 있다. 일단 전역 뷰 릴레이션이 상향식으로 구축이 되면, 관리자는 편집기를 이용하여 지역 데이터베이스와 매핑을 시행한다. 이 매핑은 스키마 매칭 기법[3, 7]과 비슷한 방법으로 수행이 되며, 단 속성과 관계를 포함한 복잡한 매칭은 제외한다.

속성의 매핑을 위한 타입을 3가지로 정하여 구현하였다.

(1) 직접 일치 : 지역 스키마의 속성 B가 전역 뷰의 속성 A와 일치할 경우

(2) 불 일치 : 지역 스키마의 속성 B가 전역스키마의 속성과 일치하는 경우가 없을 때

(3) 함수적 일치 : 전역 뷰의 속성 A가 지역 스키마의 한 개 이상의 속성과 함수적 표현(예 : 문자열 연결)으로 일치할 경우

지역 스키마와 전역 뷰의 각 릴레이션에 대하여 릴레이션의 기본 키는 릴레이션을 어떻게 매치할 것인지에 사용된다. 지역과 전역 뷰의 릴레이션을 매칭 시키는 3가지 방법이 있다.

(1) UNION : UNION을 사용하여 지역 뷰의 릴레이션들로 부터 전역 뷰에 있는 릴레이션 S를 구성한다.

(2) SINGLE : 전역 뷰의 릴레이션 S가 지역 릴레이션 1개와 일치되는 UNION의 특별한 경우이다.

(3) JOIN : 전역 릴레이션 S의 튜플들은 서로 다른 지역 릴레이션 R과 S를 조인 조건으로 연결하여 생성된다.

위의 매핑 과정들은 대부분의 일반적인 데이터

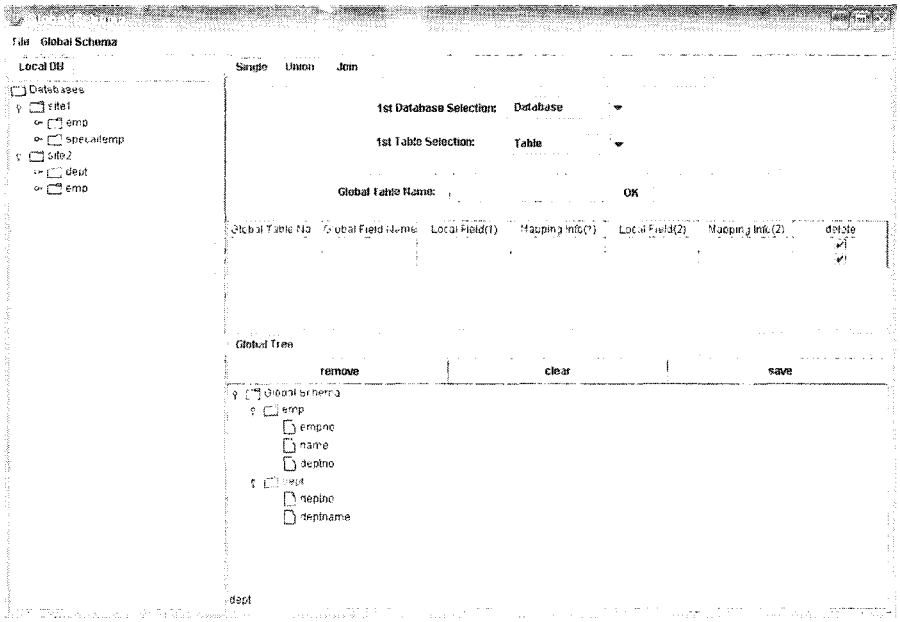
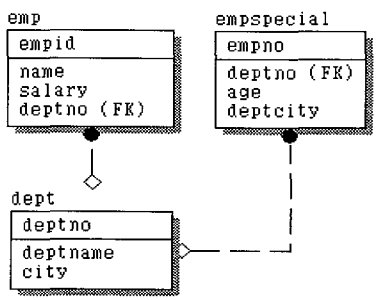



그림 4 : 전역 뷰 편집기

베이스 통합을 시행하는 경우들을 포함하며 데이터베이스 관리자나 사용자가 사용하기에 편리하도록 구성하였다.

예를 들면, 테이블 1은 그림 5의 전역 뷰에 대한 사이트 1과 사이트 2 데이터베이스의 스키마의 매핑을 보인다. 편집기는 전역 뷰 릴레이션 구축과 매핑 모두를 돕는다. 그림 4에서 지역스키마 트리는 왼쪽에 보이고 전역 뷰는 가운데 있으며 편집은 오른쪽 화면에서 이루어진다.




〔그림 5〕 전역 뷰 스키마 예

〔표 4〕 전역 뷰 스키마 매핑 예

| | Site 1 | Site 2 | Mapping |
|---|---|---|---|
| emp | db1.myemp | db2.youremp | **UNION** |
| empid | empid | empid | PK, Direct |
| name | name | name | Direct |
| salary | salary | salary | Direct |
| deptno | deptno | deptno | Direct |
| dept | | db2.deptb | **SINGLE** |
| deptno | | deptno | PK, Direct |
| deptname | | deptname | Direct |
| city | | city | Direct |
| empspecial | db1.specialemp | db2.dept | **JOIN** |
| empno | empid | | PK, Direct |
| deptno | deptno | deptno | Direct |
| age | age | | Direct |
| deptcity | | city | Direct |

전역 뷰에 대한 매핑 구축 결과는 질의처리에 사용되기 위하여 XML 매핑 파일에 저장된다. 다음 장에 보이는 GUI 질의 인터페이스는 질의 처리를 위하여 XML 매핑 파일의 정보를 사용한다. 전역 뷰를 구축하므로 써 사용자는 일반 데이터베이스 연합 시스템에서 보이는 분산 질의를 직

접 작성하는 부담을 줄일 수 있다. 본 연구의 접근법은 전역 뷰의 구축과 관리를 단순하게 하여서 데이터베이스 관리자와 사용자의 사용에 적합하다. 그림 6은 앞의 전역 스키마에 대한 XML 매핑 파일의 일부를 보인다.

```
<TABLE>
    <tableName>EMP</tableName>
    <tableType>1</tableType>
    <tableTypeName>UNION</tableTypeName>
    <FIELD>
        <fieldName>empid</fieldName>
        <MAPPING>
        <mapName>db1.myemp.empid</mapName>
        <databaseName>Site1</databaseName>
        </MAPPING>
        <MAPPING>
        <mapName>db2.youremp.empid</mapName>
        <databaseName>Site2</databaseName>
                                    </MAPPING>
    </FIELD>
    <FIELD>
        <fieldName>deptno</fieldName>
        <MAPPING>
        <mapName>db1.myemp.deptno</mapName>
        <databaseName>Site1</databaseName>
        </MAPPING>
        <MAPPING>
        ...
                (생략)
```

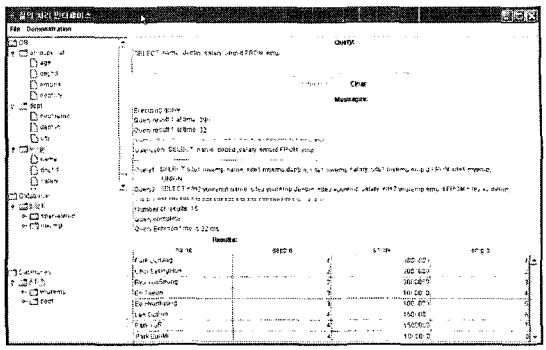
〔그림 6〕 XML 매핑 파일 예

### 3.3 전역 뷰 질의

통합 스키마에 대한 질의는 GUI 인터페이스를 통하여 이루어진다(그림 7). 사용자는 전역 뷰를 보면서 화면에 질의를 입력한다. SQL 질의는 XML 매핑 파일 정보와 실행 엔진을 가진 질의처리기에 의하여 처리된다.

전역 스키마와 지역 스키마 테이블과의 관계에 따라 질의 처리 규칙은 다음과 같은 범주로 구분된다.

(1) 타입 1 : 단일 지역 사이트에서 나오는 단순 소스 테이블을 위한 질의이다.
(2) 타입 2 : 2개 이상의 지역 사이트 테이블의 합집합으로 구성되는 질의이다.



〔그림 7〕 전역 질의처리 인터페이스

(3) 타입 3 : 2개 이상의 지역 사이트 테이블의 조인으로 구성되는 질의이다.
(4) 타입 4 : 전역 뷰에 있는 2개의 테이블을 조인하여 구성되는 질의이다.

각 질의 타입의 재구성 규칙은 다음과 같다.

(1) 타입 1 : 전역 릴레이션과 필드의 이름을 지역 릴레이션 필드와 매핑한다.
(2) 타입 2 : UNION을 이용하여 결합된 테이블을 포함한 각 소스를 지역 질의로 바꾼다. 지역 질의의 결과를 UNION 한다. UNION은 만약 두 테이블의 속성과 필드 타입이 다르면 키가 되거나 공통이 되는 속성에 대하여 적용이 된다.
(3) 타입 3 : 매핑에 정의된 키 속성들에 대하여 2개의 데이터베이스에 있는 조인 릴레이션에 대하여 연합된 질의를 생성한다.
(4) 타입 4 : 전역 뷰의 조인으 타입2의 UNION과 타입3의 연합 JOIN으로 전역 뷰의 테이블의 매핑에 따라 매핑이 된다. 다중 전역 JOIN은 연합 질의를 여러 개의 UNION과 연합 JOIN으로 바꾼다.

일단 전역 질의가 연합 질의로 번역이 되면 실행을 위하여 연합 질의 엔진에 보내진다. 다음은 앞의 예제에 대한 스키마이고 표 2는 몇 가지 예제 질의와 그 해석 결과를 보인 것이다.

[Local Schema 1 - Site 1]

myemp(empid, name, salary, deptno)

specialemp(empid, age, deptno)

[Local Schema 2 - Site 2]

youremp(empid, name, salary, deptno)

dept(deptno, deptname, city)

[Global Schema]

emp(empid, name, salary, deptid)

dept(deptno, deptname, city)

empspecial(empno, deptno, age,deptcity)

〔표 5〕 예제 질의와 질의처리기의 해석 결과

| Type 1: Single Table |
|---|
| **Global Query(전역질의):**<br>SELECT deptname<br>FROM dept<br>**Federated Query(연합질의):**<br>SELECT db2.dept.deptname<br>FROM db2.dept |
| Type 2: Union |
| **Global Query(전역질의):**<br>SELECT name<br>FROM emp<br>**Federated Query(연합질의):**<br>SELECT db1.myemp.name<br>FROM db1.myemp<br>UNION<br>SELECT db2.youremp.name<br>FROM db2.youremp |
| Type 3: Federated Join |
| **Global Query(전역질의):**<br>SELECT empspecial.empno, empspecial.deptcity<br>FROM empspecial<br>**Federated Query(연합질의):**<br>SELECT db1.specialemp.empid, db2.dept.city<br>FROM db1.specialemp, db2.dept<br>WHERE db1.specialemp.deptno = db2.dept.deptno |
| Type 4: Global Join |
| **Global Query(전역질의):**<br>SELECT emp.name, dept.deptname<br>FROM emp, dept<br>WHERE emp.deptno = dept.deptno<br>**Federated Query(연합질의):**<br>SELECT db1.myemp.name,<br>db2.dept.deptname<br>FROM db1.myemp, db2.dept<br>WHERE db1.myemp.deptno =<br>db2.dept.deptno<br>UNION<br>SELECT db2.youremp.name,<br>db2.dept.deptname<br>FROM db2.youremp, db2.dept<br>WHERE db2.youremp.deptno = db2.dept.deptno |

## 4. 시스템의 기능 비교

서론에서 설명한 바와 같이 현재도 여러 기관과 기업에서 분산된 데이터/정보에 대해서 데이터 통합이라는 과정을 추진 중이다. 근본적인 통합 데이터를 구축하기에는 비용과 기술적으로 미흡한 부분이 존재한다.

대부분의 상용 데이터베이스는 동사 제품을 통한 사적인 데이터베이스 통합 방법을 제공한다. Oracle은 데이터 마이그레이션을 통한 데이터베이스 통합을 제공하며 지역 데이터베이스의 데이터베이스 스키마와 구조를 서버로 이전한다. 그러나 이 접근법은 지역 데이터베이스 자치성을 유지하는 실시간 접근을 허용하지 않는다. 마이크로소프트의 SQL Server는 비즈니스 프로세스나 데이터 통합을 위하여 BizTalk 서버를 제공한다[9]. IBM DB2는 이질 데이터베이스에 대한 접근을 쉽게 하기 위하여 WebSphere Information Integrator를 제공한다. IBM WII는 실시간 접근을 허용하지만 DB2[10]에 많이 의존하며 데이터를 통합하는 모든 작업이 질의 재작성에 의한 수작업으로 진행된다. 상용 데이터베이스 시스템 중 전역 뷰 구축을 지원하고 질의를 허용하는 시스템은 없다.

2000년 이전의 통합 기술들은 XML 같은 메타

데이터 표준 스펙을 사용하지 않은 사적인 스펙에 따라 진행되었다. 메디에이터 구조[11,12]에 기반한 이전의 연구들은 표준 수행 엔진과 사용용이성의 부족으로 많이 보급되지 못하였다. 본 연구는 Clio[13]과 COMA++[14] 같은 스키마 매칭 시스템에서 사용하는 유사한 기법들을 사용하였으며 스키마 머지[15] 분야와 관련성을 갖는다. 본 연구의 기법은 연합 모델에 기반하지만 데이터베이스 관리자가 복잡한 연합 질의를 작성하는 것을 피하게 한다. 시스템은 전역 뷰의 구축과 뷰의 유지보수 관련 작업을 단순화하여 전역 뷰로부터 연합 질의를 자동적으로 구축하도록 하여 데이터 연합 시스템의 장점을 갖도록 지원하였다.

상업용이 아닌 연구 프로토타입 시스템으로는 TSIMMIS[11]는 미국 스탠포드 대학 Hector Garcia-Molina 교수에 의하여 개발 중인 시스템으로 Wrapper 기반의 통합 시스템이다. HERMES[1]는 Widerhold 교수에 의하여 진행된 메디에이터 기반 시스템이다. DISCO[1]는 프랑스 Valduriez 교수에 의하여 진행된 시스템이다. 정확한 비교는 어렵지만 본 시스템과 위 시스템들을 기능에 의하여 비교해보면 표 3과 같으며 기능의 정도에 따라 표시를 하였다. 본 연구에서는 뷰에서의 데이터 조작의 여러 문제점을 해소하기 위하여 질의 인터페이스에서 데이터 변경 기능은 제공하지 않는다.

〔표 3〕 기존 데이터 통합 도구 비교

| | 제안시스템 | TSIMMIS | HERMES | DISCO |
|---|---|---|---|---|
| 이질 데이터베이스에 접근 | O | △ | O | O |
| 웹과 연동 | O | X | △ | △ |
| 뷰 관리 | X | modification | modification | modification |
| 분산환경 | 이기종 시스템 | 이기종 시스템 | 단일기종 시스템 | CORBA |

## 5. 결론 및 향후 과제

데이터베이스의 증가와 비즈니스 인텔리전스 요구에 의한 통합의 필요성으로 볼 때, 데이터베이스 통합의 필요성이 증가하고 있다. 데이터베이스 통합은 세 가지 관점에서 장점, 단순화 된 시스템 관리 및 유지보수, 빠른 통합 응용 구축, 사용자에게 통합된 정보 접근이다.

본 연구에서는 데이터베이스 연합 구조의 맨 위층에 데이터베이스 통합 층을 구축하는 데이터베이스 통합 시스템을 제시하였다. 전역 뷰는 단순하고 구축이 쉽도록 하였고, 연합질의가 전역 뷰를 통하여 자동적으로 구축되도록 질의해석기를 구현하였다. 이 방법은 데이터베이스 연합의 기존 단점을 보완하며 통합 데이터에서 연합 질의를 구축하는 어려움을 없애는 장점들을 가져다 준다. 또 상용 데이터베이스의 접근 방법에 의존하지 않으므로, 데이터베이스 통합의 용이성에 비용 절감의 효과가 있다.

앞으로의 연구는 메디에이터에서 수행되는 SQL 질의를 확장하여 사용자의 편의성을 높이며, 실제 인터넷 상거래 시스템에 시험 적용을 하며, 지역 상거래 데이터베이스를 통합하여 전체 상거래 품목을 조회하는 시스템의 실험 구축을 통하여 시스템의 안정화와 유용성을 확보하고자 한다.

## 참 고 문 헌

[1] Data Integration Projects World-Wide, 2005, http://www.ifi.unizh.ch/staff/pziegler/IntegrationProjects.html.

[2] Sheth, A., Larson, J., 1990, Federated Database Systems for Managing Distributed, Heterogenous and Autonomous Databases. ACM Computing Surveys 22, 183-236.

[3] Rahm, R. and Bernstein, P., 2004, A Survey of Approaches to Automatic Schema Matching, VLDB Journal, 10 (4). 334-350.

[4] Ziegler, Patrick and Klaus R. Dittrich., 2004, Three Decades of Data Integration - All Problems Solved?, 18th IFIP World Computer Congress (WCC 2004), volume 12, Building the Information Society, Kluwer, pages 3-12, Toulouse, France, August 22-27.

[5] Mason, T. and Lawrence, R., 2005, Dynamic Database Integration in a JDBC Driver. in 7th International Conference on Enterprise Information Systems, 326-333.

[6] UNITYJDBC, 2006, http://www.unityjdbc.com/, UnityJDBC- Integrate SQL Data Sources using a Single Query.

[7] Halevy, A., 2001, Answering Queries Using Views: A survey. VLDB Journal, 10 (4). 270-294.

[8] IBM, 2005, http://www.ibm.com, A Simple Introduction to Using DB2 Information Integrator with Oracle 9i.

[9] MICROSOFT,2005, http://www.microsoft.com/korea/TechNet/biztalk/biztalka.asp

[10] Josifovski, V., Schwarz, P.M., Haas, L.M.H. and Lin, E.T., 2002, Garlic: A New Flavor of Federated Query Processing for DB2. in SIGMOD, 524-532.

[11] Chawathe, S., et al., 1994, The TSIMMIS Project: Integration of Heterogeneous Information Sources, Proceedings of the 10th Meeting of the Information Processing of Japan, pp. 7-18.

[12] Collet, C., Huhns, M., Shen, 1991, W.M.: Resource Integration Using a Large Knowledge Base in Carnot. IEEE Computer 24, 55-62.

[13] Haas, L.M., Hernández, M.A., Ho, H., Popa, L. and Roth, M., 2005, Clio Grows Up: From Research Prototype to Industrial Tool. in SIGMOD, 805-810.

[14] Aumueller, D., Do, H.H.D., Massmann, S. and Rahm, E.R., 2005, Schema and Ontology Matching with COMA++. in SIGMOD 906-908.

[15] Dragut, E. and Lawrence, R., 2004, Composing Mappings between Schemas using a Reference Ontology. in ODBASE, 783-800.

[16] Litwin, W., Mark, L. and Roussopoulos, M., 1990, Interoperability of Multiple Autonomous Databases. ACM Computing Surveys, 22 (3). 267-293.

[17] Batini, C., Lenzerini, M., Navathe, S., 1986, A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18, 323.364

[18] Ziegler, Patrick, 2004, User-Specific Semantic Integration of Heterogeneous Data: What Remains to be Done? Technical Report ifi-2004.01, Department of Informatics, University of Zurich.

[19] 박우창, 어태임, 2007, 전역 뷰를 이용한 데이터베이스 통합 시스템, 덕성여자대학교 자연과학논문집 13권, pp.101-113.

[20] 박우창, 2002, XML Schema를 이용한 이질의 데이터베이스 스키마 통합, 인터넷정보학회, 6권2호, pp.99-115.

## ◐ 저 자 소 개 ◑

**박 우 창(U-chang Park)**
1982년 서울대학교 계산통계학과 졸업(학사)
1985년 서울대학교 대학원 계산통계학과 졸업(석사)
1993년 서울대학교 대학원 계산통계학과 전산과학전공 졸업(박사)
1988~현재 덕성여자대학교 정보공학대학 교수
관심분야 : 데이터베이스, 인터넷 응용, 알고리즘
E-mail : ucpark@duksung.ac.kr