논문 2010-2-2

데이터베이스 유사도 감정 방안에 관한 연구

이주일*, 이원석**

A Study on Database Similarity Assessment Method

Jooil Lee*, Won Suk Lee**

요 약

정보 시스템의 기본적인 구성 요소인 데이터베이스의 활용이 보편화되고 고도화되면서 기존의 비즈니스 정보 관리/검색 분야에 국한되어 사용되던 관계형 데이터베이스는 90년대부터 XML 및 멀티미디어 데이터 등의 다양한 데이터 유형을 지원하고, 객체지향형 데이터 모델을 포괄적으로 수용하면서 다양한 영역에서 활용되고 있으며 많은 양의 데이터에 대한 OLAP 및 데이터 마이닝 작업을 수행하는 분석계 시스템으로 그 활용 범위가 확대되고 있다. 이에 따라서 데이터베이스의 활용 방법에 있어서도 고도의 창의력이 요구되는 부분이 증가하고 있다. 본 연구에서는 데이터베이스 분야에서 개인의 창의적 산출물로 인정할 수 있는 부분을 도출하고 이에 대한 유사도 감정 방안을 제시한다.

Abstract

The database, one of the basic component of information systems, is becoming more common nowadays and its utilization is highly enhanced. In the past, applications of relational database were confined in the field of business information management and search. Since the 1990s, it began to support various types of data such as multimedia and XML, and it adopted the objected oriented data model comprehensively. Now, the database expands its application area into the analytic systems: OLAP and data mining. For this reason, it will require us more creativity to take advantage of the database. In this study, we first define the creative output candidates in the database. Then, based on these candidates we propose the assessment method for measuring similarity between two databases.

한글키워드 : 데이터베이스 유사도, 데이터베이스 창의적 산출물, 유사도 비교 방안

*, ** 연세대학교 컴퓨터과학과
** 이원석 (교신저자)
(email: leewo@database.yonsei.ac.kr)
† 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2010-0008007)
접수일자: 2010.10.5 수정완료: 2010.11.13

Ⅰ. 서 론

컴퓨터의 발전과 더불어 인간은 일상생활에서 수많은 데이터를 접하게 되었으며, 편리한 방법

으로 데이터를 관리하는 방법을 모색하게 되었다. 이런 필요로 인해 등장한 데이터베이스는 데이터들의 모임을 의미하며 대규모 정보를 편리하게 관리하도록 설계되었다.

데이터베이스는 특정한 의미를 가지는 실세계를 반영하는 데이터들의 모임으로 체계적으로 정리하여 컴퓨터에 기억시켜 축적한 것[1][2]을 말한다. 이러한 데이터베이스는 전통적으로 문자열이나 숫자 형태의 정보를 저장하는데 사용되었으며, 최근에는 그림, 동영상, 음성, 지리 정보 등을 저장하고 검색하는데 활용되고 있다. 데이터 웨어하우스(data warehouse)와 OLAP 시스템[3][4]은 대규모 데이터베이스를 분석하여 기업의 의사결정에 사용하고 있다. 또한 실시간 데이터베이스와 능동 데이터베이스 기술은 실시간 처리나 생산과 제조 과정을 제어하는데 이용되고 있다. 한편, 데이터베이스 검색 기술은 인터넷 검색 성능을 높이게 위해 웹에도 적용되고 있다. 이처럼 데이터베이스는 점점 더 다양하고 복잡한 기능을 제공하며 많은 분야에 활용되고 있으며 그 사용법에 있어서도 고도의 창의력이 요구되는 부분이 증가하고 있다.

데이터베이스 유사도는 그 관점에 따라 다양하게 정의 될 수 있으며, 유사도 비교 방식에 따라서 그 값이 달리 계산된다. 일반적으로 데이터베이스는 데이터베이스를 프로그램의 하나로 보는 프로그램 관점과 데이터베이스를 저작물의 하나로 보는 두 가지 관점으로 볼 수 있다. 프로그램 관점은 데이터베이스 스키마의 테이블과 속성 및 함수나 저장 프로시저 이름, 질의문 중 예약어나 식별자 등의 토큰이나 스키마 제약 조건들의 구조적 특성, 그리고 함수나 저장 프로시저의 코드 특성 등을 고려한다. 컴퓨터프로그램 분쟁조정 사례[5] 중 “키워드검색프로그램 무단복제 사용에 따른 손해배상청구”는 데이터베이스를 프로그램 관점에서 본 사례로서 피신청인이 개발한 데이터베이스 검색 시스템의 독창성이 인정되어 프로그램보호법 위반이라고 주장할 수 없다는 결정이 나왔다.

반면에 저작물 관점은 데이터베이스 시스템 내에 저장된 데이터의 유사성을 고려한다. 저작권 분쟁 사례[6] 중 “데이터베이스의 무단사용에 따른 분쟁”은 데이터베이스를 저작물의 관점에서 본 사례로서 피신청인이 신청인의 맛집 GPS 데이터베이스를 복제하여 판매한 것에 대해 신청인의 권리를 침해한 것으로 인정되었다.

국내의 데이터베이스 저작권에 대한 인식은 한국저작권위원회 국내 조정통계 표[7]를 통해 볼 때 상대적으로 낮은 편이며, 컴퓨터 프로그램의 유사도의 한 측면으로 고려되는 사례가 많은 편이다. 이는 데이터베이스 유사도 비교 후보군들과 그 비교 방법에 대한 명확한 기준이 정립되지 않았다는 점을 시사한다. 전 세계적으로는 우리나라가 속한 아시아 지역이 경제 규모가 크면서도 저작권에 대한 인식 수준이 낮아 경제적으로 가장 큰 피해를 끼치는 지역으로 분류되고 있는 실정[8]이다.

이에 본 논문에서는 점차 그 활용이 다양해지고 고도화되고 있는 데이터베이스에서 개인의 창의적 산출물로 인정할 수 있는 요소를 도출하고 이에 대한 유사도 감정 방안을 제시하여 데이터베이스 저작권에 대한 인식을 높이고자 한다.

II. 데이터베이스 창의적 산출물 후보군

데이터베이스는 매우 복잡하고 다양한 요소들로 정의되며 이들 간의 유기적인 관계를 통해 운영되는 시스템이다. 이러한 데이터베이스 응용 프로그램에서의 창의적 산출물 후보군은 크게 데이터베이스 정의 및 기술과 관련된 선언부 그룹과 생성된 데이터베이스를 조작하는 조작부 그룹

의 요소들로 나눌 수 있다. 또한 저작물 관점에서 데이터베이스에 저장된 내용, 즉 데이터 자체도 후보군의 될 수 있다.

선언부 그룹은 데이터 모델, 스키마, 데이터 정의어(DDL) 등을 포함한다. 데이터 모델은 데이터베이스의 구조를 명시하기 위해 사용할 수 있는 개념들의 집합으로서 데이터 타입, 관계, 제약조건들을 포함한다. 스키마는 데이터베이스에 대한 기술로서 설계과정에서 명시하며, 주로 도식화된 표기법을 가진다. 대표적으로 관계 스키마[9]가 있다. 스키마는 DDL을 통해 정의되며, 관계 스키마에서 테이블 속성 이름이나 구조적 제약조건 등이 선언부 후보군의 예가 될 수 있다.

조작부 그룹은 데이터 조작어(DML)를 통해 생성된 데이터베이스에 대한 검색, 삽입, 삭제, 수정 연산들의 절차적 또는 비절차적 집합을 포함한다. 대화식이나 범용 프로그래밍 언어에 삽입된 질의문이나 함수 및 저장 프로시저가 조작부 후보군의 예가 될 수 있다. 또한 대량의 데이터에 대한 분석질의나 데이터마이닝 작업 연산 및 XML질의 또한 이에 포함된다.

저장 데이터는 데이터베이스의 스키마에 따라 정형화되어 저장된 데이터들의 모임을 말한다. 이중 창의적 산출물 후보군은 시스템 운영을 위해 필요한 데이터들을 배제하고 지식 서비스를 제공하기 위한 데이터들의 집합으로 한정한다.

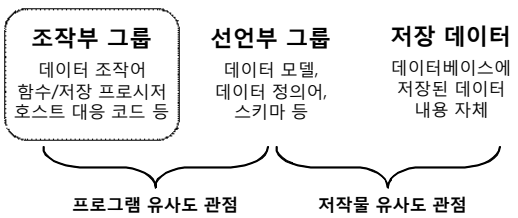

그림 1. 데이터베이스 창의적 산출물 후보군
Fig. 1. Creative output candidates in database

III. 데이터베이스 스키마

관계 모델은 데이터베이스의 개체 및 관계를 릴레이션의 모임으로 표현한다. 릴레이션은 곧 튜플과 속성으로 이루어진 테이블로 생각할 수 있으며, 속성의 데이터 값은 다수의 무결성 조건으로 제약된다.

3.1 데이터 무결성 제약조건 유사도

무결성 제약 조건은 데이터베이스 연산이 일어났을 경우 데이터의 일관성을 유지하기위해 보장해야 하는 제약으로 키, 도메인, 참조, 개체 무결성 제약 조건이 있다. 테이블이나 속성은 여러 데이터베이스마다 서로 공유되는 경우가 많은 반면 무결성 제약 조건은 데이터베이스를 활용하는 집단의 성격이나 환경 및 운영자에 따라 매우 주관적이다. 다만, 이러한 무결성 제약 조건은 테이블 및 속성의 수에 매우 의존적이므로 데이터베이스 스키마가 충분한 복잡도를 가질 때 의미 있는 유사도 측정 척도가 될 수 있다.

무결성 제약 조건의 유무 및 그 수는 데이터 선언 언어문을 통해 판별할 수 있다. 먼저 키 제약 조건은 릴레이션이 중복된 키 값을 가질 수 없는 조건으로 UNIQUE KEY 절로 판단할 수 있다. 도메인 제약 조건은 속성값의 범위를 제한하는 조건으로 CHECK 절을 통해 판단이 가능하다. 또한 참조 무결성 제약 조건은 외래키의 값이 다른 테이블의 기본키 값이거나 널 값이어야 한다는 조건으로 FOREIGN KEY 절을 조사하면 된다. 마지막으로 개체 무결성 제약 조건은 키본키 값은 널 값이 될 수 없다는 조건으로 PRIMARY KEY 절을 통해 판단할 수 있다.

데이터 무결성 제약조건들의 유사도 계산 절차는 그림 2와 같다. 제약조건을 검출한 후 각 유형별 유사도 수치를 통합하여 전체 데이터 무

결성 제약조건 간의 유사도를 계산한다. A, B를 각각 원본과 비교본이라고 할 때, 원본과 비교본의 유사도는 식 1과 같다.

$$S_{const}(A,B) = \sum_{k \in K} w_k \cdot C(A_k, B_k) \quad \cdots \text{(식 1)}$$

$K = \{u, d, r, p\}$ 이며 각각 키, 도메인, 참조, 개체 무결성 제약조건을 의미하며, w_k는 제약조건 k의 가중치이다. 가중치의 합은 1이다. C는 표 1에서 선택된 유사계수 중 하나이다.

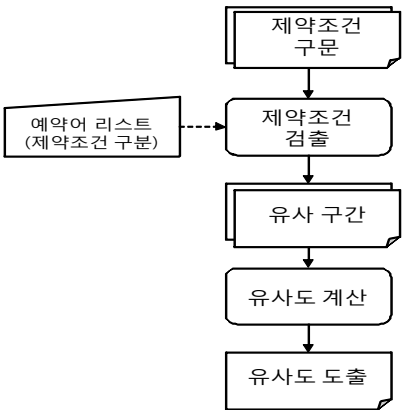

그림 2. 제약조건 유사도 계산
Fig. 2. Measuring similarity between constraints

표 1은 유사계수의 종류와 그 계산 방법이다.

표 1. 유사계수
Table 1. Similarity coefficients

유사 계수	계산식
타니모토 계수 (Tanimoto coefficient)	$\frac{N_{a\cap b}}{N_a + N_b - N_{a\cap b}}$
다이스 계수 (Dice's coefficient)	$\frac{2\lvert N_{a\cap b}\rvert}{N_a + N_b}$
자카드 계수 (Jaccard's coefficient)	$\frac{N_{a\cap b}}{N_{a\cup b}}$
코싸인 계수 (Cosine coefficient)	$\frac{N_{a\cap b}}{\sqrt{N_a} \times \sqrt{N_b}}$
중복도 계수 (Overlap coefficient)	$\frac{N_{a\cap b}}{\min(N_a, N_b)}$

3.2 데이터베이스 스키마 유사도

스키마는 데이터베이스에 대한 기술로서 설계 과정에서 명시하며, 주로 도식화된 표기법을 가진다. 일반적으로 하나의 데이터베이스는 특정 회사 또는 학교와 같이 하나의 사회 또는 군집의 단위를 포괄하여 표현할 수 있어야 하므로, 대개의 경우 데이터베이스 스키마의 수가 수십 개에 이르며 각 테이블들이 서로 연결된 형태를 띠게 되어 스키마 구조는 더욱 복잡해진다.

데이터베이스 스키마가 충분한 복잡도를 가질 때, 스키마를 구성하는 테이블 및 속성 이름, 데이터 선언 언어문의 코딩 기법 또한 유사도를 측정하는 척도로서 사용될 수 있다. 데이터 무결성 조건과 마찬가지로 데이터 선언 언어문을 통해 그 유사도를 측정할 수 있다. 다만, 이 절에서 언급한 후보군들은 데이터 무결성 조건보다 다소 주관적인 성격이 약하므로 일반적으로 사용되는 단어와 전문적이고 특수한 목적으로 사용되는 단어를 차별화하여 유사도 측정에 반영해야 한다.

데이터베이스 스키마의 유사도 계산 절차는 기본적으로 그림 2와 유사한 골격을 갖는다. 입력은 스키마 DDL이 되며, 제약 조건 검출 단계 대신 DDL 파싱과 토큰 분리 단계를 거친다. DDL 파싱 단계에서는 복잡한 스키마 DDL 구문을 복수개의 단순한 DDL 구문으로 분리한다. 토큰 분리 단계에서는 DDL 구문을 해석하여 불용어를 제거한 후 식별자, 예약어 및 특수어를 분리해 내어 식별자, 데이터 타입, 코딩 기법의 유사도를 계산할 수 있도록 준비한다.

IV. 데이터베이스 검색 및 분석

SQL(Structured Query Language)은 포괄적인 데이터베이스 언어로서 데이터베이스에서 자료

의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안된 컴퓨터 언어이다. SQL은 또한 데이터베이스로부터 정보를 얻거나 갱신하기 위한 표준 대화식 프로그래밍 언어로서 다양한 프로그램에 함께 내장되거나 데이터베이스 관리 시스템에 내장되어 존재하기도 한다. 최근에는 데이터 웨어하우징과 OLAP 그리고 XML 응용 등을 위한 선택적인 패키지가 정의되어 다양한 목적으로 활용되고 있다.

4.1 운영 및 분석질의 유사도

SQL은 크게 데이터베이스 관리 시스템을 운영하거나 검색하는 운영질의 용도, 그리고 데이터베이스 관리 시스템으로부터 특정한 목적의 분석을 수행하는 분석질의 용도로 사용된다. 질의는 그 대상 데이터베이스의 스키마를 그대로 반영하며 스키마에 따라 그 활용폭이 제한된다. 따라서 서로 다른 두 데이터베이스 관리시스템을 운영하고 검색하는 운영질의와 특정한 목적의 분석을 수행하고 결과를 도출해 내는 분석질의가 유사하다면 두 데이터베이스는 설계상으로 어느 정도 유사하다는 결론을 내릴 수 있다.

운영 및 분석질의 유사도 계산 절차는 그림 2와 유사한 골격을 갖는다. 이때 입력은 운영 및 분석질의가 되며, 제약 조건 검출 단계 대신 질의 파싱과 질의 구획화 및 토큰 분리 단계를 거친다. 질의 구획화 및 토큰 분리 단계에서는 질의 구문을 해석하여 불용어를 제거한 후 절 단위로 구분한 후 식별자, 예약어 및 특수어를 분리해 내어 식별자, 코딩 기법의 유사도를 계산할 수 있도록 준비한다.

한편 SQL은 다양한 언어로 작성된 수많은 응용프로그램에 내장되어 프로그램의 각 기능 단위별로 하나 또는 그 이상의 절차적 질의와 연계되어 프로그램을 통해 미리 작성된 SQL 문이 수행되는 방식으로 많이 사용될 수도 있다. 따라서 응용 프로그램 상에 기능 별로 내장된 질의문이 서로 유사한지를 비교 판단한다면 이를 통해서도 데이터베이스의 유사도를 유추해 낼 수 있다.

4.2 저장 프로시저

저장 프로시저(stored procedure)[10]는 일련의 검색, 삽입, 삭제, 수정 연산 및 프로그래밍 요소들의 집합을 데이터베이스에 미리 저장해 둔 것을 말한다. 저장 프로시저는 데이터베이스 관리 시스템 내의 다른 저장 프로시저를 통해 호출될 수도 있으나, 일반적으로 호스트 프로그램 내의 대응하는 코드를 통해서 호출되고 결과를 돌려주는 방식으로 주로 사용된다.

저장 프로시저와 호스트 대응 코드는 프로그램 코드와 그 구조가 흡사하기에 고도의 창의력을 요하며, 작성자의 코딩 스타일이나 그 성향을 잘 반영하는 특징을 갖는다. 따라서 저장 프로시저의 유사도는 저장 프로시저의 이름, 파라미터와 같은 식별자의 유사도와 저장 프로시저에 포함된 질의 문들의 유사도, 그리고 표 2와 같은 소프트웨어 유사도 판단 방법을 적용한 유사도를 종합함으로써 계산할 수 있다. 식별자 유사도와 질의문의 유사도는 각각 3.2절과 4.1절의 유사도 계산 절차를 참고하여 계산할 수 있다.

표 2. 소프트웨어 유사도 판단 방법
Table 2. Method to measure software similarity

판단방법	유 형
지문법 방식	• 주석의 삽입, 삭제 등 편집 • 함수/변수의 이름 및 타입, 값 변경
구조적 방식	• 불필요한 코드의 삽입 • 선언된 함수/변수의 순서 변경 • 다른 코드들의 조합 • 함수/모듈을 분할하여 코드 생성 • 함수/모듈을 통합하여 코드 생성

Ⅴ. 데이터베이스 저장 데이터

데이터베이스에 저장된 데이터의 유사도는 데이터베이스의 내용적인 유사도를 의미한다. 저장 데이터의 유사도를 산출하기 위해서는 먼저 데이터베이스에 저장된 내용을 기능적으로 나누는 작업이 선행되어야 한다. 일반적으로 데이터베이스는 정규화 된 수많은 테이블로 존재하므로 원본과 비교본에서 동일한 기능을 하는 테이블이라 하더라도 정규화 정도에 따라서 서로 다른 복수개의 테이블로 존재하게 된다. 이 경우 유사도 값을 산출해 내기가 매우 어렵게 된다. 따라서 기능적으로 분류된 각각의 테이블 집합에 대해서 역정규화를 수행하여 이러한 문제점을 최소화해야 한다. 그림 3은 테이블 비정규화 과정을 보여준다.

그림 3에서 원본과 비교본 각각 기준 테이블과 이 테이블과 외래키 관계에 있는 모든 테이블을 하나의 큰 테이블로 만들고, 이렇게 역정규화된 테이블 간 저장된 데이터 내용을 비교하게 된다. 이때, 비교하는 테이블 간 공통되지 않는 속성의 데이터 내용은 비교 대상에서 배제한다.

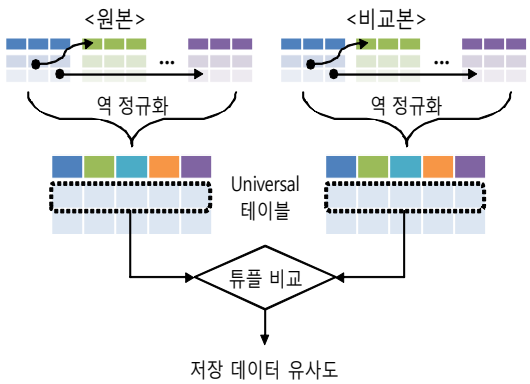

그림 3. 저장 데이터 유사도 비교
Fig. 3. Measuring similarity of contents

Ⅵ. 유사도 비교 방안

두 데이터베이스의 유사여부를 판단할 때, 두 데이터베이스 중 어느 데이터베이스를 기준으로 유사도를 산출하느냐에 따라 그 결과가 달라질 수 있다. 원본 기준 방식은 비교본이 원본의 일부 핵심 부분만을 복제한 경우에는 불리하다. 반면에 비교본 기준 방식은 비교본이 독자적으로 추가한 부분이 많을수록 유사도가 낮다. 또한, 유사도 계산방식은 기준영역을 선정하는 방식에 따라 전체영역 기준 방식과 공통영역 기준 방식으로 나눌 수 있다.

일반적인 경우 복제된 데이터베이스는 원본에서 핵심적이거나, 개발에 많은 시간 및 인력이 소요되는 것, 또는 독창적인 아이디어가 포함된 부분을 복제한 후 여기에 새로운 기능을 추가하기 위하여 여러 가지 수정이나 보완을 가함으로써 복제 사실을 은폐하려는 경향을 갖는다. 그러므로 이 경우 복제된 데이터베이스에서 수정 및 추가된 부분의 많고 적음은 유사도 판단의 중요한 기준이 되지 않는다.

6.1 유사도 유형

지금까지 살펴본 데이터베이스 창의적 산출물 후보군들은 그 특징 및 유사도 비교의 공통적인 성격에 따라 표 3과 같이 나눌 수 있다.

표 3. 유사도 유형
Table 3. Type of similarity

유 형		포함 후보군
스키마 유사도	토 큰 유사도	식별자, 예약어
	구 조 유사도	무결성 제약조건
코드 유사도		질의, 함수, 저장 프로시저, 호스트 대응 코드
데이터 유사도		저장 데이터

토큰 유사도는 데이터베이스 스키마의 테이블과 속성 또는 함수나 저장 프로시저 이름, 질의문 중 예약어나 식별자의 유사도를 말하며, 이들 간의 일치도에 따라 계산한다. 이때, 이름이나 식별자의 리터럴 값이 일반적으로 많이 사용되는 값인지, 특수한 기관이나 조직에서만 사용될 수 있는지 여부에 따라 차등을 두어 가중치를 부여해야 한다. A, B를 각각 원본과 비교본이라고 하고 w_i의 가중치를 가지는 k_T개의 토큰이 존재할 때, 토큰 유사도는 식 2와 같다. 단, 가중치의 합은 1이다.

$$S_T(A,B) = \sum_{i=1}^{k_T} w_i \cdot C(A_i, B_i) \quad \cdots \text{(식 2)}$$

구조 유사도는 데이터베이스 스키마의 구조의 유사도를 의미한다. 구체적으로는 키 무결성 조건, 도메인 무결성 조건, 참조 무결성 조건, 개체 무결성 제약 조건 및 기타 제약 조건에 대한 유사도를 말하며, 이들 간의 유사도는 식 3과 같이 계산한다.

$$S_S(A,B) = S_{const}(A,B) \qquad \cdots \text{(식 3)}$$

코드 유사도는 데이터베이스 운영 및 분석 질의 문과 절차식 질의의 경우 질의의 순서, 함수 및 저장 프로시저의 구조 및 기능의 유사도를 의미한다. 추가적으로 저장 프로시가 존재할 경우 이와 대응되는 호스트 언어 대응 코드의 유사도 또한 포괄적으로 의미한다.

코드 유사도는 표 2의 소프트웨어 유사도 판단 기준을 적용하여 그 유사도를 종합적으로 계산하는 것이 바람직하다. 또한 코드 개발자의 숙련도에 따라 데이터베이스 복제 수준이 달라질 수 있다. 예를 들면, 숙련도가 낮은 개발자의 경우 코드를 그대로 복제하는 경우가 있는 반면에 높은 숙련도를 가진 개발자의 경우 식별자의 리터럴 값이나 코드의 구조를 교묘하게 변경할 확률이 높다. 따라서, 코드 유사도 비교 시 개발자의 숙련도를 반영하는 것이 바람직하다.

P를 소프트웨어 유사계수라 하고 w_i의 가중치를 가지는 k_C개의 개체가 존재할 때, 코드 유사도는 식 4와 같다. 단, 가중치의 합은 1이다.

$$S_C(A,B) = \sum_{i=1}^{k_C} w_i \cdot P(A_i, B_i) \quad \cdots \text{(식 4)}$$

데이터 유사도는 데이터베이스에 저장된 데이터의 유사도를 의미한다. 데이터 유사도를 계산하기 위해서는 먼저 원본과 비교본에서 기능적으로 유사한 테이블 집합에 대해서 그림 3과 같이 역정규화를 수행해야 한다.

w_i의 가중치를 가지는 k_D개의 개체가 존재할 때, 데이터 유사도는 식 5와 같다. 단, 가중치의 합은 1이다.

$$S_D(A,B) = t \times \sum_{i=1}^{k_D} w_i \cdot C(A_i, B_i) \quad \cdots \text{(식 5)}$$

실제로 비교 가능한 테이블의 수 k_D는 모든 가능한 테이블의 수에 비해 적은 정도가 달라질 수 있다. 따라서, 유사도 계수에 반영 비율 t를 곱한다.

6.2 최종 유사도 산출

이전 절에서 정의한 데이터베이스 유사도 유형을 바탕으로 각 유형별로 가중치를 주거나 유사도 관점을 달리해서 최종적인 유사도를 산출해 낼 수 있다. W_M , W_C , W_D를 각각 스키마 유사도 가중치, 코드 유사도 가중치, 데이터 유사도 가중치라고 할 때, 프로그램 관점과 저작물 관점에서의 최종 유사도는 식 6과 같이 계산할 수 있다. 단, $S_M(A,B) = S_T(A,B) + S_S(A,B)$이고, 각 식에서 가중치의 합은 1이다.

$$S_{Prog}(A,B) = W_M \cdot S_M(A,B) + W_C \cdot S_C(A,B)$$
$$S_{Cont}(A,B) = W_M \cdot S_M(A,B) + W_D \cdot S_D(A,B)$$
… (식 6)

Ⅶ. 결 론

정보 시스템의 기본적인 구성 요소인 데이터베이스의 활용이 보편화되고 고도화되면서 개인의 창의력이 요구되는 부분이 증가하고 있다. 따라서, 데이터베이스에 대한 개인의 창의력을 인정하고 그 저작권을 보호하는 방안에 대한 필요성이 증대되고 있다. 이러한 데이터베이스의 저작권은 데이터베이스 설계를 기술하는 스키마와 데이터베이스를 활용하는 질의를 분석하고 비교함으로서 가능하다. 무결성 제약 조건과 스키마의 특징 비교, 운영질의와 분석질의 및 저장 프로시저 등이 유사도를 판별하는 기본적인 후보로서 이용될 수 있다. 데이터베이스의 유사도는 스키마 유사도, 구조 유사도, 데이터 유사도 유형의 합으로 표현될 수 있으며, 유사도 관점과 유사도 산출 방식에 따라 각 유사도 유형의 가중치를 조절하여 데이터베이스의 전체 유사도를 유연하게 계산할 수 있다.

참 고 문 헌

[1] R. Elmasri and S. Navathe, “Fundamentals of Database Systems”, Addison-Wesly, 2007

[2] A. Silberschatz, H. Korth, and S. Sudarshan, “Database System Concepts (4/E)″, McGraw-Hill, 2001

[3] Bill Inmon, “Building the Data Warehouse″, John Wiley & Sons, 1996

[4] Ralph Kimball and Margy Ross, “The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2/E)″, John Wiley & Sons, 2002

[5] 한국저작권위원회, “저작권분쟁조정사례 제3편 컴퓨터프로그램 조정사례”, 2010

[6] 한국저작권위원회, “저작권분쟁조정사례 제2편 일반저작물 분쟁 사례”, 2010

[7] 한국저작원위원회, “조정통계(2010년 10월 기준)”, 2010

[8] Business Software Alliance and IDC, “09 Piracy Study″, 7th Annual BSA/IDC Global Software, May 2010

[9] C. J. Date, ″SQL and Relational Theory″, O′Reilly, 2009

[10] B. Pribyl and S. Feuerstein, “Oracle PL/SQL Programming (5/E)″, O′Reilly, 2009

저 자 소 개

이주일 (Jooil Lee)

2006년 2월: 홍익대학교 컴퓨터공학 공학 석사

2009～현재: 연세대학교 컴퓨터과학과 박사과정

<주관심분야 : Database, Data Stream Mining, Context Awareness>

이원석 (Won Suk Lee)

1985년 7월 : Boston University 컴퓨터과학 공학 학사

1987년 7월: Purdue University 컴퓨터과학 공학 석사

1990년 7월: Purdue University 컴퓨터과학 공학 박사

2004년～현재: 연세대학교 컴퓨터과학과 정교수

<주관심분야 : Sensor Data Stream Processing, Data Stream Mining, OLAP / Data Warehouse>