

SPARQL-to-Original SQL 변환 모델

성하정*, 김장원*, 이석훈*, 백두권**

*고려대학교 컴퓨터·전파·통신공학과

e-mail : octoom@korea.ac.kr, ikaros1223@korea.ac.kr, leha82@korea.ac.kr, baikdk@korea.ac.kr

A Model for SPARQL-to-Original SQL Translation

Hajung Sung*, Jangwon Gim*, Sukhoon Lee*, Doo-Kwon Baik**

*Dept. of Computer and Radio Communications Engineering, Korea University

요 약

시맨틱 웹 환경을 통해 수 많은 온톨로지가 생성되고 이를 효율적으로 관리하고 저장하기 위한 온톨로지 저장소에 대한 연구가 진행되고 있다. 더불어 저장된 온톨로지로부터 원하는 정보를 얻기 위해 SPARQL 에 대한 연구 또한 활발히 진행되었다. 현재까지 많은 데이터들이 관계형 데이터베이스에 저장되어 있기 때문에 SPARQL 을 사용하기 위해서는 해당 저장소의 모델에 맞게 SPARQL 을 변환 해야 한다. 하지만 지금까지 연구된 SPARQL-to-SQL 변환 모델은 관계형 데이터베이스에 저장된 데이터를 트리플 형태로 변환하여 온톨로지 저장소에 저장해야 하는 추가 비용이 발생한다. 이 논문에서는 관계형 데이터베이스에 저장된 데이터를 트리플 형태로 변환하여 온톨로지 저장소에 저장하지 않고 SPARQL 질의문의 변환만으로 관계형 데이터베이스에 저장된 데이터에 질의 할 수 있는 SPARQL-to-Original SQL 변환 모델을 제안한다.

1. 서론

시맨틱 웹 기술이 발전함에 따라 지식을 표현 하기 위한 온톨로지 사용이 증가 하고 있다. 이를 저장하고 관리하기 위한 온톨로지 저장소에 대한 연구가 활발히 진행 되고 있다. 대표 적인 온톨로지 저장소는 Sesame, Jena, Oracle, SDB, DLDB, OWLJessKB 등이 있으며, 이들은 관계형 데이터 베이스를 기반으로 온톨로지를 저장한다[1-6]. 온톨로지 저장소에서 원하는 온톨로지를 획득하기 위해 SPARQL 을 사용한다[7]. 그러므로 SPARQL 을 통해 관계형 데이터베이스 기반의 온톨로지 저장소로부터 온톨로지를 가져오기 위해서 SPARQL 은 해당 저장소의 모델에 맞게 변환되어야 한다[8-14]. SPARQL-to-SQL 변환 모델들은 특정 저장소의 구조에 종속적일 뿐만 아니라, 일반적인 관계형 데이터베이스에 저장된 데이터들에는 질의를 할 수 없는 문제를 가진다. 그 결과 SPARQL 질의를 수행하기 위해서는 관계형 데이터베이스에 저장되어 있는 데이터 들을 온톨로지 저장소에 트리플 형태로 변환시켜 저장해야 하므로 추가적인 변환 비용이 발생하게 된다[15]. 이러한 문제를 해결하기 위해 데이터의 변환 없이 질의 문의 변환 만으로 원하는 결과를 가져올 수 있는 SPARQL 질의문에 대한 변환 모델이 필요하다. 이 논문에서는 SPARQL 을 관계형 데이터베이스에 질의 할 수 있는 SPARQL-to-Original SQL 변환 모델을 제안 한다.

이 논문의 구성은 다음과 같다. 제 2 장에서는 이 논문의 관련 연구를 기술하며, 제 3 장에서는 SPARQL to Original SQL 변환 모델을 제안하며, 제안 모델을 기반으로 한 구현 결과를 4 장에서 설명한다. 마지막으로 5 장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련연구

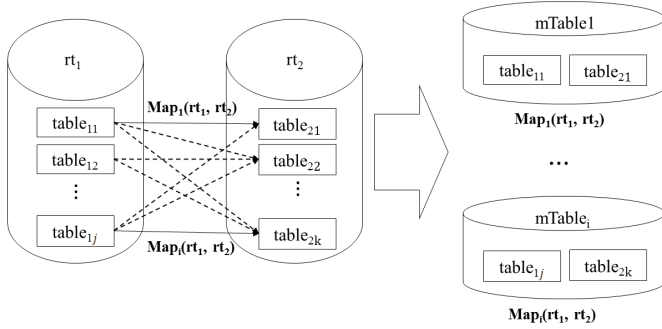
온톨로지 저장소에 저장되어 있는 데이터를 가져오기 위한 연구들로, D2RQ, ORACLE, R2RML 등이 있다. D2RQ 는 관계형 데이터베이스에 저장된 데이터를 매핑 파일을 이용하여 RDF 형태의 온톨로지로 변환한다. 변환 된 온톨로지는 메모리에 저장되어 추가적인 온톨로지 저장소를 구성한다. D2RQ 는 메모리에 저장된 온톨로지를 통해 온톨로지 질의를 지원한다. 위와 같은 작업은 대용량의 데이터를 온톨로지로 변환하는데 적합하지 않다. 또한 D2RQ 가 변환한 온톨로지는 관계형 데이터베이스 데이터와 실시간으로 동기화 되지 않는다. 즉 관계형 데이터베이스에 저장된 데이터들이 변경될 경우 이를 즉시 반영 할 수 없다. Oracle 11g Release2 는 Semantic Network 모듈을 통해 트리플 형식으로 저장된 데이터에 대한 온톨로지 질의를 지원한다. 이를 위해서는 Oracle 내부에 RDF 형식의 온톨로지 저장소를 추가로 생성해야 한다. R2RML 은 W3C 에서 개발하고 있는 RDB-to-RDF 매핑 언어로서, 관계형 데이터베이스와 온톨로지의 매핑관계를 정의하기 위한 규칙 및 방법을 제공한다. 하지만 현재까지 R2RML 을 기반으로 하는 시스템이 존재하지 않는다.

이 연구에 참여한 연구자는 '2 단계 BK21 사업' 의 지원을 받았음

† : 교신저자

또한, SPAQRL 을 SQL 로 변환하는 알고리즘이 개발되었다. 그렇지만 변환된 SQL 은 트리플 기반의 온톨로지 저장소에서만 적용이 가능하다. 그러므로 일반적인 관계형 데이터베이스를 위한 SQL 로의 매핑이 필요하다. 이 논문에서는 일반적인 관계형 데이터베이스에 존재하는 데이터에 SPAQRL 질의가 가능하도록 SPARQL-to-Original SQL 변환 모델을 제안한다.

3. 제안 모델



(그림 1) SPARQL-to-Original SQL 매핑 모델

그림 1 은 SPARQL-to-Original SQL(StO) 매핑 모델을 나타낸다. SPAQRL 을 Original SQL 로 변환하기 위해 SPAQRL 과 SQL 의 매핑은 그림 1 의 매핑 방법을 따른다. 매핑은 첫째 SPARQL 에서 사용한 트리플 집합 (T)를 구성한다. 둘째 트리플 집합을 이용하여 트리플과 관계형 데이터 베이스 스키마와 매핑되는 관계형 트리플 집합(RT)을 구성한다. 마지막으로 관계형 트리플 집합을 이용하여 관계형 데이터베이스의 매핑 테이블 집합($MTABLE$)을 구성한다.

첫 번째 단계인 트리플 집합을 구성하는 방법은 SPARQL 에서 정의된 트리플의 Subject, Predicate, Object 속성을 사용 한다. 두 번째 단계는 첫 번째 단계에서 정의한 트리플 속성을 이용하여 트리플과 관계형 데이터베이스 스키마를 매핑한다. 매핑된 정보는 각 트리플에 매핑 된 관계형 데이터 베이스를 테이블 정보를 가지고 있으며 이를 관계형 트리플을 집합이라 한다. 세 번째 단계는 관계형 트리플의 속성들에 대해 데카르트 곱(Cartesian product) 연산을 실시하여 새로운 매핑 테이블 집합을 구성한다. $mTable_i$ 는 매핑 테이블 집합에 속하는 i 번째 매핑 테이블 집합이다. 그림 1 은 SPARQL 에서 두 개의 트리플을 질의한다고 가정한다. rt_1 에 j 개의 테이블이 존재하고 rt_2 에 k 개의 테이블이 존재 한다고 가정하면 $j \times k$ 개의 조합이 가능하다. $j \times k$ 를 i 로 정의 하여 i 개의 조합이 가능하다고 가정 하였다. $mTable_i$ 의 첫 번째 속성 $table_{1j}$ 를 rt_1 과 두 번째 속성 $table_{2k}$ 는 rt_2 와 매핑 된다. i 개의 조합에 대한 SQL 문을 완성 후 Union 연산을 통해 SPARQL 에서 Original SQL 변환을 완료한다. 매핑 모델을 위해 필요한 정의는 다음과 같다.

(정의 1) $T = \{t | t \text{ 는 SPARQL 이 표현 하는 모든 트리플}\}$
 $t_i = \{s_i, p_i, o_i\}$

모든 트리플의 집합을 T 라 하고 i 번째 트리플을 t_i 로 정의한다. t_i 의 속성은 subject, predicate, object 이며 각각 s_i, p_i, o_i 로 나타낸다.

(정의 2) $RT = \{rt | rt \text{ 는 } t \text{ 가 표현 하는 모든 테이블}\}$
 $rt_i = \{table_{i1}, table_{i2}, \dots, table_{ij}\}$

트리플의 패턴은 관계형 데이터베이스의 테이블과 매핑 작업을 정의 한다. 즉 각각의 트리플 패턴 정의와 일치하는 관계형 데이터베이스 테이블을 모두 찾아서 집합으로 구성하여 관계형 트리플(RT)이라 정의한다. rt_i 는 관계형 트리플의 i 번째 트리플이며 속성은 rt_i 와 트리플 패턴과 일치하는 관계형 데이터베이스의 테이블 집합을 속성으로 갖는다.

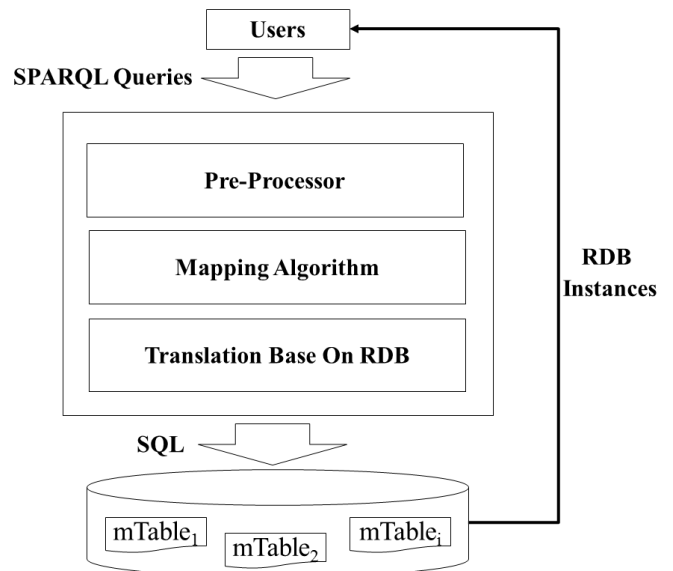
(정의 3) $MTABLE = rt_1 \times rt_2 \times \dots \times rt_i$
 $mTable_i \in MTABLE$

이를 위해 데카르트 곱(Cartesian product) 연산을 적용 하였다.

$$rt_1 \times rt_2 \times \dots \times rt_i = \{(table_1, table_2, \dots, table_i) | table_1 \in rt_1, table_2 \in rt_2, \dots, table_i \in rt_i\}$$

관계형 트리플은 SPARQL 에서 질의 하는 트리플 수와 일치한 i 개의 rt 를 속성으로 갖는다. i 개의 rt 간 데카르트 곱 집합을 매핑 테이블 집합으로 정의한다. 매핑 테이블 집합은 모든 rt 의 속성 수의 곱만큼 $mTable$ 을 속성으로 갖는다.

4. 구현



(그림 2) SPARQL-to-Original SQL 아키텍처

그림 2 는 SPAQRL 을 Original SQL 로 변환하기 위한 구현 시스템의 아키텍처를 보인다. 첫 번째 단계인 전 처리기는 사용자에게 입력 받은 SPARQL 을 트리플 저장소에 질의 가능한 SQL(T-SQL)문으로 변환한다. 두 번째 단계인 RDB to Triple Mapping 은

R2RML 에서 제안한 매핑 방법을 기반으로 관계형 데이터베이스 스키마를 매핑한다. 매핑 파일은 XML 으로 구현하였으며 R2RML 에서 제안한 LogicalTable 개념을 이용하여 관계형 데이터베이스의 테이블과 매핑 시킨다. 다음으로 SubjectMap 과 PredicateObjectMap 은 각각 주 키 정보와 테이블 열 명 정보와 매핑 시킨다. 세 번째 단계는 T-SQL 문을 Original SQL(O-SQL)문으로 변환한다. 변환 하는 방법은 T-SQL 에서 트리플을 정의 하고 있는 Selection 부분을 참고하여 트리플 집합을 구성한다. 트리플 집합은 정의 2,3 에 따라 관계형 트리플 집합, 매핑 테이블 집합의 순으로 변환 된다.

매핑 테이블 집합의 구성이 완료되면 T-SQL 을 네 가지 단계로 나눠 분석하여 O-SQL 의 변환을 완료한다. 첫 번째 단계는 Projection 에 대한 변환으로 아래와 같이 변환한다.

- Subject 는 매핑 되는 테이블의 주 키에 대한 열 이름으로 변환한다.
- Object 는 트리플이 명시한 열 이름으로 변환한다.

위에 처리 방법에 따라 변환이 완료 되면 ($t_i.subject$) 문장은 ($t_i.주 키에 대한 열 이름$)으로 변환된다. 두 번째 단계는 Selection 에 대한 변환으로 아래와 같이 변환한다.

- Subject 는 주 키에 대한 열 이름으로 변환한다.
- Object 는 트리플의 정의 된 열 이름으로 변환한다.

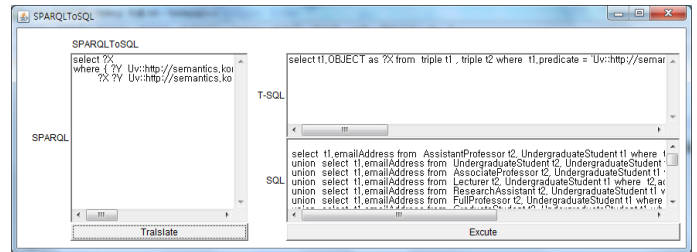
위에 처리 방법에 따라 변환이 완료 되면 ($triple_i.subject = value_x$) 문장은 ($triple_i.주 키의 열 이름 = value_i$) 로 변환된다. 세 번째 단계는 Join 에 대한 변환으로 아래와 같이 변환한다.

- Subject 는 주 키의 열 이름으로 변환된다.
- Object 는 트리플에 정의 된 열 이름으로 변환한다.

위에 처리 방법에 따라 변환이 완료 되면 ($triple_i.subject = triple_j.object$) 문장은 ($triple_i.주 키의 열 이름 = triple_j.열이름$) 으로 변환된다. 네 번째 단계는 From 구문에 대한 변환으로 매핑 테이블 집합의 속성으로 각 트리플을 치환 한다. 위에 처리 방법에 따라 변환이 완료 되면 $mTable_i$ 의 속성 $table_{1j}, table_{2k}$ 가 From 구문 내 $triple_1, triple_2$ 의 문장을 치환한다. 따라서 ($triple_1, triple_2$) 문장은 ($table_{1j}, table_{2k}$)으로 변환 된다. 치환 과정은 매핑 테이블의 원소 수만큼의 진행된다. 그 결과 매핑 테이블의 원소 수와 같은 SQL 문이 생성되며 각 SQL 문을 Union 연산자로 연결하여 O-SQL 변환 작업을 완료한다.

그림 3 은 제안 모델을 구현한 것으로 SPARQL 을 입력하면 SPARQL 을 T-SQL 문으로 변경하고, 이를 O-SQL 변환한다. 그 결과 변환된 쿼리를 실행시켜 원

하는 결과를 가져온다. 구현된 모델을 실험하기 위해 온톨로지 생성도구 인 UBA(Univ-Benchmark Artificial Data Generator)를 이용하여 온톨로지 데이터를 생성하였다[16]. 실험을 위해 관계형 데이터베이스에도 UBA로부터 생성되는 정보를 동일하게 생성하였다. 표 1 은 SPARQL-to-Original SQL 변환 결과를 보여준다. Q_i 은 사용자가 입력한 SPARQL 질의어이며 전 처리기에 의해 트리플 저장소에 질의 가능한 T-SQL 인 T_i 으로 변환된다. T_i 을 O-SQL 인 O_i 으로 변환하여 SPARQL-to-Original SQL 변환을 완료한다.



(그림 3) SPARQL to Original SQL 구현 화면

<표 1> 질의 어 변환 결과

구분	질의 어
Q_i	select ?X where { ?Y Uv::http://semantics.korea.ac.kr/~ikaros/univ-bench.owl#emailAddress ?X ?Y Uv::http://semantics.korea.ac.kr/~ikaros/univ-bench.owl#advisor Uv::http://www.Department5.University0.edu/AssistantProfessor7}
T_i	select t1.object AS X from Triple t1, Triple t2 where t1.predicate = 'Uv::http://semantics.korea.ac.kr/~ikaros/univ-bench.owl#emailAddress' and t2.predicate = 'Uv::http://semantics.korea.ac.kr/~ikaros/univ-bench.owl#advisor' and t2.object = Uv::http://www.Department1.University0.edu/AssistantProfessor7 and t1.subject = t2.subject
O_i	select t1.emailAddress from AssistantProfessor t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from UndergraduateStudent t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from AssociateProfessor t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from Lecturer t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from ResearchAssistant t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from FullProfessor t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from GraduateStudent t1, UndergraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress from AssistantProfessor t1, ResearchAssistant t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name union select t1.emailAddress

from UndergraduateStudent t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from AssociateProfessor t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from Lecturer t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from ResearchAssistant t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from FullProfessor t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from GraduateStudent t1, ResearchAssistant t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from AssistantProfessor t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from UndergraduateStudent t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from AssociateProfessor t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from Lecturer t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from ResearchAssistant t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from FullProfessor t1, GraduateStudent t2
where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name
union select t1.emailAddress
from GraduateStudent t1, GraduateStudent t2 where t2.advisor = 'AssistantProfessor7' and t1.name = t2.name

5. 결론

이 논문은 관계형 데이터베이스 데이터를 온톨로지 형태로 변환하지 않고 SPAQRL 을 관계형 데이터베이스에 질의 할 수 있는 SPARQL-to-Original SQL 변환 모델을 제안한다. 이를 위해 R2RML 에서 제안한 매핑 모델을 적용하였으며, 실제 구현을 통하여 관계형 데이터베이스로부터 SPARQL 질의가 요청하는 결과를 가져올 수 있었다. 이 논문에서는 SPARQL 의 selection, projection, inner join 연산만을 고려하였다. 그렇지만 기존의 연구들이 가지는 트리플 저장소로의 데이터의 변환 비용보다 효율적으로 원하는 결과를 가져올 수 있음을 보였다. 또한 관계형 데이터베이스의 정보가 변경될 경우에도 변경된 사항이 실시간으로 반영되어 정보의 누락을 방지할 수 있다. 향후 연구로서, StO 매핑 모델과 기존 연구에 대한 정량적인 비교 평가가 필요하며 다양한 관계형 데이터 베이스에 적용하기 위해 SPARQL 의 모든 문법을 지원하는 것이 필요하다.

참고문헌

[1]Jeen Broekstra, Arjohn Kampman, Frank van Harmelen, "Sesame: A Generic Architecture for Storing and

- Querying RDF and RDF Schema," Springer-Verlag, Lecture Notes in Computer Science (LNCS), Vol.LNCS 2342, pp. 54-68, June 2002.
- [2]Jena - A Semantic Web Framework for Java. <http://jena.sourceforge.net>
- [3] http://download.oracle.com/otndocs/tech/semantic_web/pdf/oradb_semantic_overview.pdf
- [4]<http://jena.hpl.hp.com/wiki/SDB>
- [5]Pan, Z. and Heflin J., "DLDB: Extending Relational Databases to Support Semantic Web Queries," InWorkshop on Practical and Scaleable Semantic Web Systems, The Second International Semantic Web conference (ISWC2003), 2003.
- [6]OWLJessKB : A Semantic Web Reasoning Tool, <http://edge.cs.drexel.edu/assemblies/software/owljesskb>
- [7]SPARQL Query Language for RDF, W3C, 4 October 2006.
- [8]A. Chebotko, S. Lu, H.M. Jamil, F. Fotouhi, Semantics preserving SPARQL-to-SQL query translation for optional graph patterns, Technical Report TR-DB-052006-CLJF, Wayne State University, May 2006. <<http://www.cs.wayne.edu/~artem/main/research/TRDB-052006-CLJF.pdf>>.
- [9]R. Cyganiak, A relational algebra for SPARQL, Technical Report HPL-2005-170, Hewlett-Packard Laboratories, 2005. <<http://www.hpl.hp.com/techreports/2005/HPL-2005-170.html>>.
- [10]S. Harris, N. Shadbolt, SPARQL query processing with conventional relational database systems, in:Proc. of the International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS), pp. 235- 244, 2005.
- [11]sparql2sql – a query engine for SPARQL over Jena triple stores, <http://jena.sourceforge.net/sparql2sql>
- [12]Yannis Theoharis, Vassilis Christophides, and Grigoris Karvounarakis, "Benchmarking database representations of RDF/S stores," Springer-Verlag, Lecture Notes in Computer Science (LNCS), Vol. LNCS3729, pp. 85-701, 2005.
- [13]Dongwon Jeong, Myounghoi Choi, Yang-Seung Jeon, Youn-Hee Han, Laurence T. Yang, Young-Sik Jeong, and Sung-Kook Han, "Persistent Storage System for Efficient Management of OWL Web Ontology," Springer-Verlag, Lecture Notes in Computer Science (LNCS), Vol. LNCS 4611, pp.1089-1097, July 2007.
- [14]손지성, 정동원, 백두권 “SPARQL-to-SQL 변환 알고리즘의 저장소 독립적 활용을 위한 시스템모델”, 한국정보과학회지, 제 14 권, 제 5 호, pp.467-471, 2008년 7월
- [15]S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T.Thibodeau, S. Auer, and J. Sequeda, A. Ezzat, A Survey of Current Approaches for Mapping of Relational Databases to RDF," Technical Report,January 2009 (available at http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf).
- [16] Guo, Y., Pan, Z., and Heflin. J., "An Evaluation of Knowledge Base Systems for Large OWL Datasets," Springer Verlag, Third International Semantic Web Conference, Vol. LNCS 3298, pp. 274-288,2004.