통합 데이터베이스 스키마 검사기의 설계와 구현

김규백[O] 유경용 김형찬
삼성전자 디지털솔루션센터
{gyubaek.kim[O], kyongyong.yu, hckim96}@samsung.com

A Design and Implementation of Integrated Database Schema Checker

G.B. Kim[O], K.Y. Yu, H.C. Kim
Digital Solution Center of Samsung Electronics

요 약

프로젝트의 경쟁력과 성공을 위해 통합 데이터베이스 관리가 중요해지고 있다. 본 논문에서는 통합 데이터베이스 관리가 스키마 설계에서부터 이루어지도록 지원하는 새로운 스키마 검사기의 설계와 구현 내용을 소개한다. 개발된 스키마 검사기는 데이터베이스 객체의 명명 규칙 검사, 적합한 단어 필터, 유사 객체 찾기, 사용자 지정 규칙 적용의 기능을 가지고 있다. 그리고 일반적인 컴파일러의 구문 분석 과정과 다른 방법을 적용해 작업 효율을 높인 구현 세부 사항에 대해서도 상세히 설명한다. 개발된 도구는 통합 데이터베이스 관리 업무에 중요하게 현재 활용되고 있다.

1. 서 론

통합 데이터베이스는 설계에서 개발까지 전 과정이 일관성 있는 기준으로 통합되어 있음을 의미하는 것으로, 스키마의 통합, 컴포넌트들의 공유, 데이터웨어하우징, CRM(Customer Relationship Management), 데이터마이닝의 구축 비용 절감과 효과 극대화를 위한 작업들을 중심으로 이루어져 있다.

다양한 서비스의 통합이 이루어져야 하는 차세대 솔루션 개발에 있어 데이터베이스 수준에서부터의 통합 작업은 중복 방지와 효율성 제고를 통해 프로젝트의 경쟁력을 결정짓는 매우 중요한 요소이다[1].

이 중에서 데이터베이스 스키마의 통합은 가장 선행되어야 하는 것으로, 가장 적은 비용으로 큰 효과를 얻을 수 있는 개발 단계이다. 또한, 협업(Collaboration) 규모가 확대되면서 상호 업무 파악과 분석, 조정 등이 힘들어지고 있기 때문에 스키마의 통합은 필수적이다.

따라서, 개발 조직이 공유할 수 있는 스키마 설계 원칙과 명명 규칙(Naming Rule)을 검사하고 조정하는 작업을 통해 스키마 통합을 이루도록 지원하기 위한 통합 데이터베이스 스키마 검사 도구가 필요하다.

본 논문을 통해 제시하는 스키마 검사기는 데이터베이스 객체의 명명 규칙 검사, 적합한 단어 필터, 객체명과 타입을 통한 유사 객체 찾기, 사용자 지정 규칙 적용 등의 기능을 가지고 통합 데이터베이스 구축 업무에 유용하게 활용된다.

본 논문의 구성은 다음과 같다. 제 2장에서 관련된 사례 조사와 시스템 목표를 설명하고, 제 3장에서 구현 방법과 내용을 소개한다. 그리고 제 4장에서 결론을 맺는다.

2. 시스템 목표

2.1. 사례 조사

Telelogic사의 Logiscope[2]와 같은 소프트웨어 품질 검사 제품들에는 프로그래밍 코딩 규칙 검사 기능이 있으나, 데이터베이스 스키마에 대한 구문 검사에는 사용할 수 없다. 그리고, 대표적인 데이터베이스 모델링 도구인 Computer Associates사의 ERWin[3]에는 Naming Standards라는 기능이 있는데, 템플릿 형태의 제한된 명명 규칙만을 지원하고 검사할 수 있어서 각 프로젝트와 조직의 특성을 반영하는 명명 규칙을 적용하는 것 자체가 불가능해 유연성이 없다.

명명 규칙은 스키마 통합을 이루기 위한 수단으로서 상당히 중요한데, 사례들을 통해 살펴 본 일반적인 명명 규칙 구성 요소들은 다음과 같다[4,5,6].

- 대소문자 구성(Capitalization) - 파스칼(Pascal), 낙타(Camel), 전부 대문자(UpperCase)
- 접두사/접미사 및 특정 단어 포함
- 분류 방법 및 체계 적용
- 길이 제한
- 복합 단어 구성 시 개수 제한
- 다른 객체명 필수 참조
- 특수 문자 사용

하지만 이러한 명명 규칙을 모두 적용해 검사할 수 있는 제품이나 솔루션은 현재 없다. 따라서, 위와 같은 요소들에 대한 검사 기능을 가지는 도구의 개발이 필요하다. 단순히 명명 규칙의 적용 뿐만 아니라 통합 데이터 구축 관점에 필요한 다양한 기능에 대한 추가 개발도 필요하다.

2.2. 구현 목표

스키마 설계 단계를 통합적으로 관리하기 위해서 필요한 것은 다음과 같다. 첫째, 명명규칙 준수 여부 확인. 둘째, 권고 데이터 타입 사용 여부 검사. 셋째, 단어적 통일성을 유도. 넷째, 사용자 지정 규칙 적용.

이와 같은 기능 구현을 통해 상호 데이터베이스 객체들에 대한 이해를 높여 프로젝트 품질을 향상 시키고, 형태와 내용측면에서의 검사 결과에 따라 테이블과 컬럼의 중복 요소가 제거되 도록 하여 하드웨어 비용 가운데 가장 큰 비중을 차지하는 저장 장

치에 대한 낭비 요소를 없애는 것을 목표로 한다. 부수적으로 스키마 통합 작업은 데이터웨어하우징, CRM, 데이터마이닝에서 보다 효과적인 결과를 얻기 위한 정리(Cleansing) 작업의 역할도 한다.

3. 시스템 구현

3.1. 최소 구문 분석

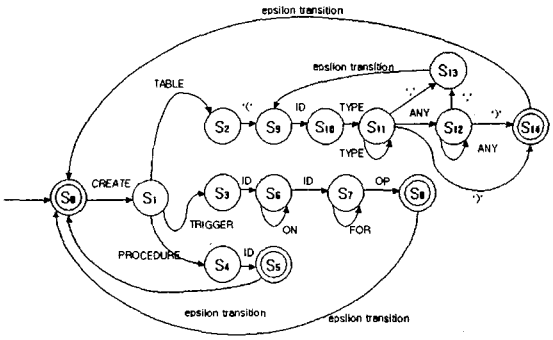
개발된 스키마 검사기는 [표 1]에서와 같이 일반적인 컴파일러의 구문 분석 과정과는 다르다.

	일반 컴파일러의 구문 분석	
토큰의 정의	구문 결정에 의미 있는 요소	
어휘 분석	키워드, 상수, 아이디로 분류	
심볼 테이블	생성함	
구문 오류	분석함	
분석 범위	전체	

[표 1] 일반적인 컴파일러 구문 분석 과정과의 비교

개발된 스키마 검사기는 라인 프로세싱을 기본으로 하여 각 라인을 토큰 스트림으로 분리 획득한다. 이와 같은 방법을 취하는 이유는 개발된 도구의 입력 파일의 SQL 문장은 이미 문법적으로 완전하고, 일정한 형태로 출력되어 있기 때문에 이를 최대한 이용하는 것이다.

또한 SQL에 대한 전체 구문을 분석 대상으로 하지 않는다. 왜냐하면 개발된 도구에서 스키마 통합의 대상은 데이터베이스 객체명과 타입 뿐이기 때문이다. 따라서, 최소 관심있는 토큰만을 구문 분석하기 위한 상태 천이 다이어그램(State Transition Diagram)이 아래와 같이 도출된다.



[그림 1] 최소 토큰 구문 분석을 위한 상태 천이 다이어그램

[그림 1]에서 ID는 테이블명, 컬럼명, 프로시져명, 트리거명을 나타내고, OP는 INSERT, DELETE, UPDATE를 나타낸다. 그리고 TYPE은 컬럼과 관련된 모든 토큰들을 나타낸다. 만약, 예로 decimal(2,3)과 같은 컬럼 타입이 여러 개의 토큰으로 떨어져 있다고 할 때, 이를 모두 획득하기까지는 S_{11} 상태에 머무른다. 끝으로 ANY는 데이터베이스 객체명 획득에 관련 없는 나머지 SQL 문법을 구성하는 토큰들이다.

초기 상태(S_0) 진입 여부를 가리기 위해 모든 토큰에 대해 이루어져야 하는 문자열 비교를 줄여 성능을 개선하기 위해 다음과 같은 필터 방법을 적용하였는데, 그 효과는 다음과 같다.

[방법] 키워드의 첫번째 문자와 토큰의 첫번째 문자 일치하지 않을 경우 무시함.

[효과] 예제 테스트 스키마에서 ERwin과 MS SQL Server에서 생성한 DDL(Data Definition Language) 파일을 각각 입력한 경우 양쪽 모두 약 75% 이상을 필터함.

3.2. 명명 규칙 검사

개발된 스키마 분석기는 앞서 2.1에서 살펴 본 일반적인 명명 규칙을 형태적으로 검사할 수 있는 메소드들을 모두 가지고 있다.

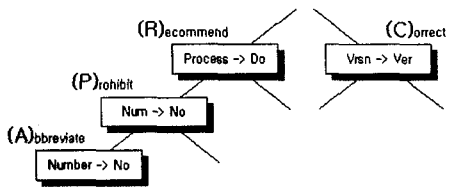
본 논문에서의 스키마 검사기는 명명 규칙 검사 기능 외에도 통합 데이터베이스 관리를 위해 필요한 다른 중요한 기능도 구현하고 있다.

3.3. 단어 필터

데이터베이스 객체에 대한 형태적인 무결성 중에 최소 의미 단위인 단어를 추출할 수 있는데, 이들에 대해 단어 필터 기능을 수행한다. 예를 들어, tShCm_EventDefine 이라는 테이블은 접두사(t), 분류체계(대분류:Sh, 중분류:Cm), 특수문자(_), 단어들(Event, Define)로 나누어져서 명명 규칙의 형태에 따라 무결한지를 검사 받는데, 이 중간 과정에서 단어들은 별도의 단어 필터 과정으로 넘어 간다.

이처럼 필요에 의해서 데이터베이스 객체 정의에 사용되는 단어들은 필터될 필요가 있는데, 이를 위해 개발된 스키마 검사기는 검사를 시작하기 전에 미리 정의된 단어 사전을 텍스트 파일로 입력 받아 이진 검색 트리를 구성한다. 단어 필터 기능은 [그림 2]의 예와 같이 각각 활용된다.

- 적절한 약어로 대체 (A)
- 잘못된 약어 교정 (C)
- 특정 단어 사용 금지 (P)
- 같은 의미의 여러 단어 가운데 하나로 권고 (R)



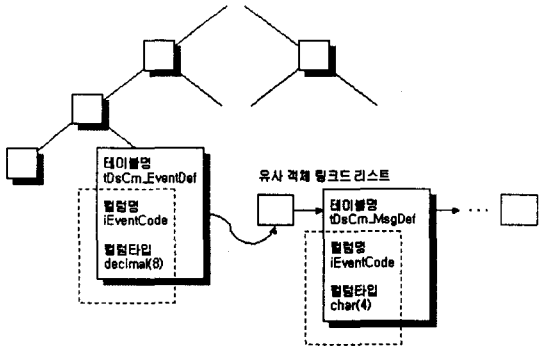
[그림 2] 단어 필터를 위한 트리 자료 구조

[그림 2]에서 Vrsn이 Ver로 필터되는 것은 (A)와 (C)가 동시에 적용된 예이다. 단어 필터 기능은 이처럼 잘 정의된 사전을 통해 적용된다.

3.4. 유사 객체 찾기

만약 어느 테이블에서 사용자 아이디 기록을 위한 타입의 길이가 다른 쪽 테이블의 경우와 다르다면 이는 오류일 가능성도 있고 일관성 차원에서도 문제가 된다. 따라서 비록 테이블은 다르지만 동일한 이름의 컬럼일 경우 데이터 타입이나 길이가 다르면 사용자에게 이것이 의도된 것인지 아닌지에 대한 경고를 나타내 준다. 이 과정을 통해 유사 가능성이 있는 객체들이 제시된다. 이들은 통합의 대상으로 고려되어야 한다.

이를 구현하기 위해서 모든 테이블의 컬럼들에 대해 {테이블명, 컬럼명, 컬럼타입}을 트리 노드의 내용으로 하는 이진 검색 트리를 구성한다. 단어 필터에 사용된 트리와는 다르게 컬럼명이 같은 다른 테이블의 컬럼들은 별도의 노드로 입력되는 것이 아니라 검색 성능 향상을 위해 [그림 3]와 같이 링크드 리스트로 유지된다.



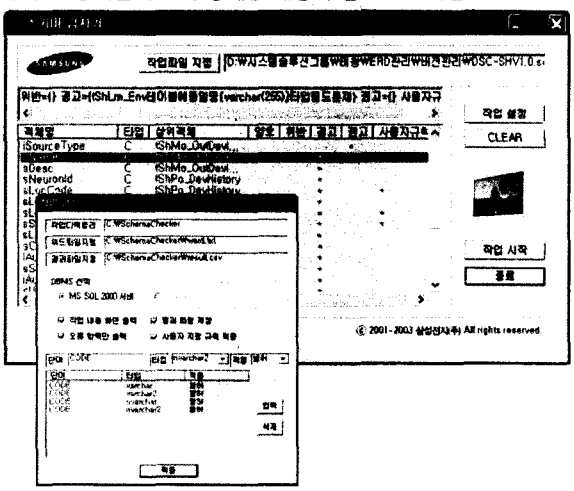
[그림 3] 유사 객체 찾기를 위한 트리 자료 구조

3.5. 사용자 지정 규칙 적용

의미적으로 어떤 용도의 컬럼은 반드시 특정 형태의 타입을 가져야 되거나, 반대로 가지면 안 되는 경우가 있을 수 있다. (예, 코드성의 컬럼 타입에는 가변 문자형을 허용하지 않음.) 이를 위해 사용자 지정 규칙을 적용한다. 이 기능의 구현을 위해서도 3.3에서 사용된 것과 같은 이진 검색 트리가 사용된다. 3.3과 3.4, 3.5의 기능 구현을 위해 사용된 이진 검색 트리는 실제는 하나의 템플릿 클래스로 구현되어 있다. 한번의 통합 스키마 검사 과정에는 내부적으로 세 개의 런타임 트리 자료 구조 생성이 포함된다.

3.6. 동작

개발된 스키마 검사기는 ANSI SQL 92 표준을 따르는 ASCII 형태의 DDL 문장들로 구성된 파일을 입력으로 한다. 또한, Microsoft SQL Server의 Generate SQL Script 기능을 통해 생성되는 T-SQL의 DDL 문장들에 대해서도 동작한다. 그리고, 결과로서 각 데이터베이스 객체에 대해 양호, 위반, 경고, 권고, 사용자 지정 규칙 준수 여부와 메세지를 출력한다. 출력은 [그림 4]에 보는 화면과 .csv 파일 형태로 나타난다. 결과 리스트에서 각 항목을 누르면 상세한 내용이 별도로 표시된다.



[그림 4] 수행 화면

테이블과 스토어드 프로시져가 각각 약 100개 컬럼 개수가 500개 이상인 예제 테스트 스키마를 2.3GHz CPU 1개, 512MB 메모리 사양의 PC에서 구동했을 때, 화면 출력까지 포함해 1초 정도의 동작 완료 시간을 보였고, 10MB 정도의 물리적 메모리 사용율과 최고 25%의 CPU 사용율을 나타냈다.

현재 Windows 하드웨어 프로파일 정보와 MD5 [7]모듈을 이용해 사이트 라이센스를 적용해 회사 내부에서만 사용하는 정책으로 모든 데이터베이스 관련 개발자들이 사용하고 있다.

개발된 스키마 검사기는 Visual Studio .NET 2003으로 개발되어, 모든 Windows 플랫폼에서 구동된다.

4. 결론

데이터베이스 설계 및 개발 과정이 후반부로 갈수록 각 데이터베이스 요소들에 대한 통합 작업은 시간과 비용을 발생시키고 어려워진다. 통합 데이터베이스 관리 시작의 시점은 따라서 스키마 설계 과정부터이어야 한다. 하지만, 프로젝트의 협업 규모가 커짐에 따라 스키마의 통합은 쉽지 않다. 이러한 문제점을 인식하고 설계, 개발되어 본 논문을 통해 제시한 통합 데이터베이스 스키마 검사기는 관련 개발자들에 자체적으로 사전에 중복 요소를 점검하는데 큰 도움이 될 뿐만 아니라, 프로젝트안에서의 상호 연관성 이해를 높여 데이터베이스 요소들이 통합 관리될 수 있도록 하는데 유용한 도구가 된다.

향후 개발된 도구를 통해 통합된 데이터베이스에서 데이터마이닝과 같은 예측 작업을 수행했을 때, 스키마의 통합 결과가 예측 결과에 얼마나 구체적으로 기여하는지에 대한 정량적인 결과 수치를 얻기 위한 연구로 발전시킬 계획이다.

5. 참고 문헌

[1] 김기윤, 노재범, " 산업지도를 바꾸는 인터넷 비즈니스" 삼성경제연구소 CEO Information, 1999년 5월

[2] Telelogic, " Logiscope - Detects Coding Error" , http://www.telelogic.com, Technical Manual, 2003

[3] Computer Associates, " AllFusion ERwin Data Modeler - Getting Started" , http://support.cai.com, Technical Manual, 2002

[4] M. Zahn, " Naming Conventions for .NET / C# Projects" , Akadia AG, March 2003

[5] ISO 11179, " Information Technology - Specification and Standardization of Data Elements"

[6] D. Georgopulos, " Develop a Consistent Naming Convention for Your Database Objects" , DevX Article, February 10, 2003

[7] R. Rivest, " The MD5 Message-Digest Algorithm", RFC 1321, April 1992