

데이터베이스 응용 소프트웨어 개발 지원 도구에 관한 연구

김아영*, 윤청*

*충남대학교 컴퓨터공학과

e-mail:sagp5456@cnu.ac.kr

A Study on the Support Tool for Database Application Software Development

Ah-Young Kim*, Cheong Youn*

*Dept of Computer Engineering, Chungnam National University

요 약

소프트웨어 산업의 발전에 따라 개발 수요의 증속 및 개발 효율성의 증대를 위해 다양한 개발 방법론과 개발 지원 도구들이 등장해 왔다. 그러나 데이터베이스 응용 소프트웨어(Database Application Software, DB-App)의 개발이 활발함에도 불구하고, 해당 분야 개발의 효율을 도모하는 방안에 관한 연구는 미비한 실정이다. DB-App의 소스 코드(Source code)에는 쿼리(Query)문이 빈번하게 사용되지만 작성 과정에서 오류가 발생하기가 쉽고 관리에 대한 어려움이 있다. 이는 개발 효율을 저하시켜 비용을 증가시키는 요인이 된다. 본 논문에서는 다양한 데이터베이스와 프로그래밍 언어로 개발되는 DB-App 개발을 지원하기 위한 방법을 기술하였으며, 사용자의 설정에 따라 쿼리문을 포함한 소스 코드를 생성하는 도구를 설계하였다.

1. 서론

정보화 시대로 들어서면서 많은 정보를 처리하고 관리하기 위해 데이터베이스(Database, DB)의 활용이 높아지고 있으며, 소프트웨어(Software, SW) 개발 프로젝트(Project)에서 DB가 제외된 경우는 거의 드문 추세이다. 흔히 DB를 구축하고 활용하기 위해 SQL(Structured Query Language, SQL)이 사용되는데, DB를 활용하는 소프트웨어 개발에서 SQL을 이용한 코드 작성은 대부분 수작업으로 이루어지게 된다. SQL을 이용한 코드를 삽입시키는 작업은 때로 매우 번거로운 작업이 되기도 하며, 코드의 실행 이전에는 문법적인 오류를 발견하지 못하는 경우가 많아 개발의 효율성을 저하시키는 원인이 된다.

현재까지 SW 개발의 생산성, 효율성 및 재사용성을 향상시키기 위한 다양한 개발 지원 도구가 등장했지만, 데이터베이스 응용 소프트웨어 개발을 지원하기 위한 도구의 개발은 미비한 상황이다. 본 논문에서는 중복될 수 있는 구문 작성 작업이 최소화될 수 있도록 자주 사용되는 SQL의 질의를 분석하였으며, 이와 관련된 소스 코드(Source Code)를 자동으로 생성하는 방법을 제시한다. 이로써 문법적인 오류가 감소되고, 실제 코드 작성에 투자되는 시간이 단축될 것으로 예상된다.

2. 소프트웨어 개발 지원 도구의 종류

소프트웨어 개발 지원 도구란 개발의 생산성과 신뢰성

등을 높이기 위해 설계자 및 개발자에게 다방면으로 편의를 제공하는 것이다. 여기에는 디버깅 도구, 형상 및 버전 관리 등의 다양한 종류의 도구가 있지만 본 논문과 비교적 유사하다고 판단되는 기준으로 서술하고자 한다.

먼저 소프트웨어 역공학(Software Reverse Engineering, SRE)이란 요구사항 정의, 분석, 설계, 구현 단계를 거꾸로 거슬러 올라가는 것을 의미하는데, 이는 원시 코드를 통해 문서와 모듈 다이어그램 등을 추출한다. 게다가 분석 단계에서 기존 A 언어로 작성된 코드를 B 언어로 변환하는 기능까지 제공한다. SRE가 원시 코드로부터 시작되는 반면, 소스 코드 자동 생성을 위한 과정은 기존 순공학의 단계로 시작된다. 이는 설계 단계에서 모듈 명세서, 자료 흐름도, 구조도 및 개체관계도 등의 자료가 입력되면 전체적인 흐름에 따라서 소스 코드를 생성하는 것이다. 마지막으로 네트워크 통신으로 복잡하게 얽힌 데이터 구조를 서버에 저장하여, 네트워크 구조의 소스 코드를 자동 생성하는 도구가 있다. 이는 알고리즘을 구현하는 사람과 이외의 시스템을 구현하는 사람으로 구분하여 두 업무 간의 상호 의존성을 낮추는데 의미를 두었다.

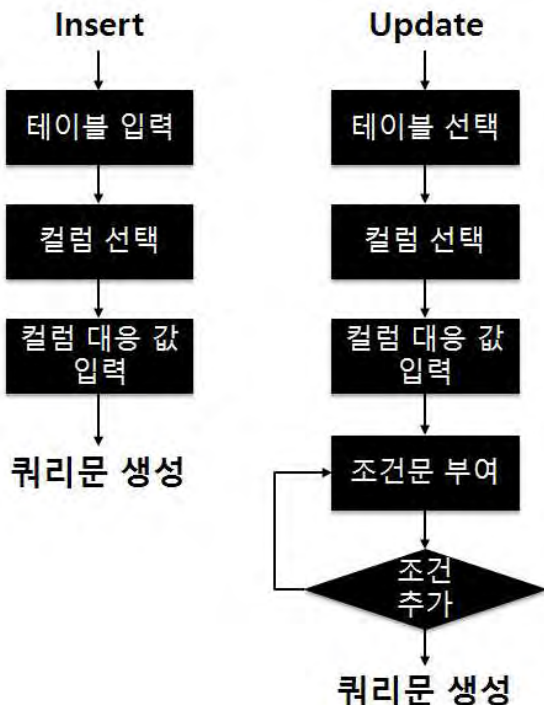
앞서 언급된 도구들은 설계 단계에서 필요한 문서들을 관리하기도 하고 프로그램의 전체적인 소스 코드의 틀을 잡아주어 코드 작성에 드는 비용을 상당 부분 감소시킨다. 이와 달리, 본 논문에서는 DB란 특정 주제를 잡아 세부적으로 도움을 주는 방안이 초점을 맞추었다.

일반적으로 DB를 쉽게 사용하기 위해 엑셀을 활용하거나 반복되는 구문을 하드 코딩으로 작성하는 경우가 많다. 그러나 일일이 SQL 질의를 작성하는 것은 문법적 지식을 요구하며, 오타로 인한 오류의 발생 가능성을 증가시키는 요인이 되기도 한다. 필요에 따라 특정 질의의 유효성을 검증하기 위해 데이터베이스 관리 시스템(Database Management System, DBMS)에서 지원되는 도구를 통해 작성한 질의를 매번 확인하기도 한다.

본 논문에서는 이러한 번거로움을 없애기 위해 자주 사용되는 질의 위주로 매크로를 정의하였으며, 이에 대한 소스 코드를 자동으로 생성하는 방안을 제시한다.

3. 매크로 정의

SQL 문법에서 주로 Insert, Select, Update 등이 빈번히 사용되는 구문이다. 구현 방식은 물론 쉬울 수 있지만 동일한 작업이 반복된다보면 개발의 효율성이 떨어지기 마련이며, 문법적 오류 또한 발생할 가능성도 높다. 게다가 Join이나 데이터 조작 함수와 같은 문법은 고도의 지식을 요하기도 한다. 이와 같이 자주 사용되거나 어려운 구문에 대해 매크로를 통한 쿼리(Query)문 자동 생성 기능을 지원하고자 한다. 그림 1과 같이, 여러 문법에 따른 각 매크로 순서도를 그려보면 서로 다른 기능이지만 비교적 유사한 순서를 갖는 것을 알 수 있다.



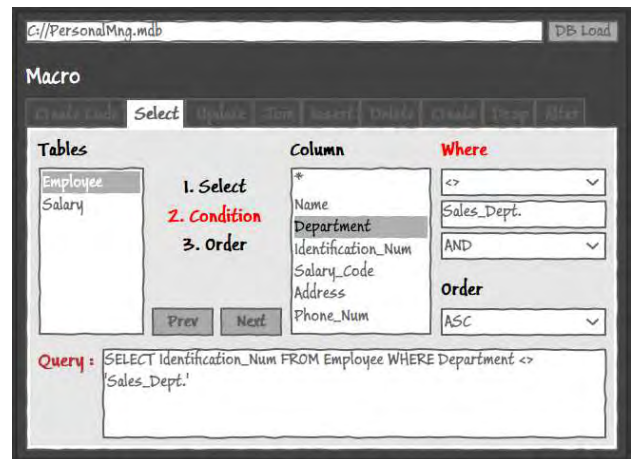
(그림 1) Insert, Update 문의 매크로 순서도

예를 들어, Insert와 Update 구문의 기본 문법은 다음과 같다. “INSERT INTO table_name (column_name1 [, column_name2, column_name3...]) VALUES (value1 [,

value2, value3...)”, “UPDATE table_name SET column_name1 = value [, column_name2 = value ...] [WHERE condition]” 두 가지 구문은 서로 다른 명령으로 시작하지만 기본적으로 테이블명, 추가/수정을 원하는 컬럼명과 이에 대응되는 값을 작성하는 등의 흐름적인 부분은 거의 유사하다. 예를 든 두 가지 구문뿐만 아니라 다른 구문들도 유사하게 적용되기 때문에, 구문별로 매크로 순서를 정의해 두어 개발의 효율성을 증대시키고자 한다.

4. 구현 방안

기본적인 흐름은 위에서 정의된 매크로 순서에 따라서 구현될 것이다. 다음 그림 2와 같이, 먼저 DB 원본이 선택되었다는 가정 하에 진행된다. DB 원본을 로드하면 우선적으로 테이블과 각 테이블에 대한 컬럼 정보를 저장한다. 다음으로 사용자가 원하는 매크로 탭을 선택하면 테이블 목록에서 하나의 테이블과 그에 해당하는 컬럼 중에서 검색할 컬럼을 선택한다. 완전한 질의 생성되면 즉시 탭 하단의 쿼리 박스에 쿼리문이 표시된다. 추가적으로 조건을 지정할 경우, 원하는 조건 개수만큼 선택이 반복된다. 마지막으로 정렬까지 입력을 해주면 사용자가 원하는 쿼리문이 완성된다. 예를 들어, “Select * From Employee Where Name = ‘홍길동’”의 구문을 작성하고자 한다면, 기존과 같이 문자열 입력이 아닌 단지 선택만 필요할 뿐이다. 이는 프로그래밍 중간에 생기는 필요한 쿼리문을 쉽게 생성해줄과 동시에 문법을 검증할 수 있게 된다. 뿐만 아니라 소스 코드를 자동 생성해 주는 부분에서 프로그램 내에서 쿼리문을 호출할 수 있도록 함수를 함께 생성해 주어 개발의 효율성을 증가시킨다.



(그림 2) 매크로 GUI 설계안

위에서 언급된 것처럼 사용자의 설정에 따라 쿼리문을 포함한 테이블의 클래스나 구조체를 자동 생성할 수 있다. 유사한 예로는 MFC 언어에서 OleDb와 연동하여 해당 클래스를 자동 생성되는 기능이다. 이는 개발에 있어서 효율성을 높일 수 있는 기능임에도 불구하고 해당 기능을 제

공하는 언어는 제한적이다. 이를 위해 문법 구조가 유사한 객체지향언어인 C++, C#, Java 언어에 대하여 우선적으로 소스 코드 자동 생성 기능을 지원할 예정이다. 그림 3과 같이, 사용자는 테이블을 먼저 선택하고 해당 테이블의 컬럼 중에서 Getter/Setter 함수를 추가할 컬럼을 체크한다. 그런 뒤에 파일 저장 경로와 추출될 개발 언어를 선택하면 파일이 생성되고 프로그램 내에서도 결과창에 전시된다.



(그림 3) 코드 생성 GUI 설계안

참고문헌

[1] Jin, Y. B., Wang, C. J. "A Study on Construction of Reverse Engineering Environment in Forth Generation Language" 1995, Korea Information Processing Society Review

[2] Lee, D. H. "Development of a Software Development Supporting Tool for Domestic Computers" 1994, Report of Electronics and Telecommunications Research Institute

[3] Park, J. Y., Kawk, D. G., Ko, B. W., Yoo, C. W. "Design Method for Language Translation" 2008, Korea Computer Congress

[4] Lee, Y. J., Kim, A. Y., Park, S. K., Oh, J. Y., Kim, J. S. "A Study on the Support Tool for Simulator Algorithm Development" 2014, Korean Institute of Navigation and Port Research Journal

5. 향후 연구 방향

DBMS에는 PostgreSQL, MySQL, MS-SQL, Oracle 등의 다양한 종류가 있다. 시스템에 따라 SQL 문법의 기본은 거의 유사하지만 구현 방식에 분명한 차이가 있다. 예를 들어, PostgreSQL에는 기존 시스템에서 볼 수 없는 배열 타입을 지원하고 있다. 본 논문에서는 현재 이러한 사항은 배제한 상태로 설계했기 때문에 이에 대한 방안을 검토할 필요가 있다.

또한 대규모 소프트웨어 개발 프로젝트에서는 소프트웨어 개발 지원 도구가 원활히 사용되고 있지만 여전히 대부분의 소규모 소프트웨어 개발 프로젝트에서는 비용적인 측면과 복잡하고 익숙하지 않다는 이유로 활용되지 못하고 있다. 이는 개발자 입장에서 정확한 니즈를 파악하지 못한 것이 이유 중에 하나일 것으로 판단된다. 차후 연구에서는 이러한 문제점들을 체계적으로 분석하여 현실적이고 실용적인 지원 도구를 구현할 예정이다.