

데이터 분석을 위한 다양한 질의를 지원하는 GUI기반 DB 액세스 인터페이스 설계

양현식, 백나은, 배유림, 장재우
 전북대학교 IT정보공학과
 e-mail:guslr1222@jbnu.ac.kr

Design of GUI based on DB Access Interface supporting various Queries for Data

Hyeonsik Yang, Naeun Baek, Yulim Bae, Jaewoo Chang*
 Dept. of Information Technology, Chonbuk Univ.

요 약

빅데이터 시대가 도래함에 따라, 방대한 양의 데이터 분석을 위한 데이터베이스 관리시스템의 중요성이 높아지고 있다. 일반 사용자는, 대표적인 데이터베이스 질의 문법인 SQL을 알지 못하기 때문에, GUI 기반 DB 액세스 툴을 사용해야 한다. 그러나 현재 개발된 GUI 기반 데이터베이스 관리 시스템은 제한적인 기능만을 제공하며, 복잡한 질의를 위해서는 사용자가 SQL 기반의 질의를 직접 입력해야 하는 문제점이 존재한다. 따라서 본 논문에서는, 데이터 분석을 위한 다양한 질의를 지원하는 GUI 기반 DB 액세스 인터페이스를 제안한다.

1. 서론

최근 SNS가 보급화되고 인터넷에 유통되는 데이터의 트래픽이 크게 증가함에 따라 빅데이터 시대가 도래하였다 [1]. 따라서 경영, 통계, 연구 등 많은 분야에서 빅데이터 분석 및 처리 서비스가 각광 받고 있다[2-4]. 기존에 데이터 관리를 위해 사용했던 스프레드시트와 같은 문서 도구는 방대한 양의 데이터처리에는 적합하지 않기 때문에[5], 일반 사용자도 효율적으로 빅데이터를 처리하기 위한 데이터베이스 관리 시스템의 중요성이 높아지고 있다. 대표적인 DB 매니지먼트 툴인 Navicat[6], Sequel Prof[7], HeidiSQL[8]을 포함한 GUI 기반 DB 액세스 툴은 기초적인 질의 수행과 관리를 GUI를 통해 지원해주지만, 여전히 복잡하고 자세한 질의문은 SQL 문법을 사용해야 하며 이는 SQL 등 전문적인 지식 없는 사용자의 질의 편의성을 크게 떨어뜨린다.

따라서 본 논문에서는 SQL 등 전문적인 지식이 없는 사용자도 GUI 기반으로 데이터에 접근하여 질의를 수행할 수 있는 DB 액세스 인터페이스 및 데이터 처리 기법을 제안한다. 제안하는 기법에서 지원하는 질의는 SELECT, JOIN, SUB-QUERYING 질의이다.

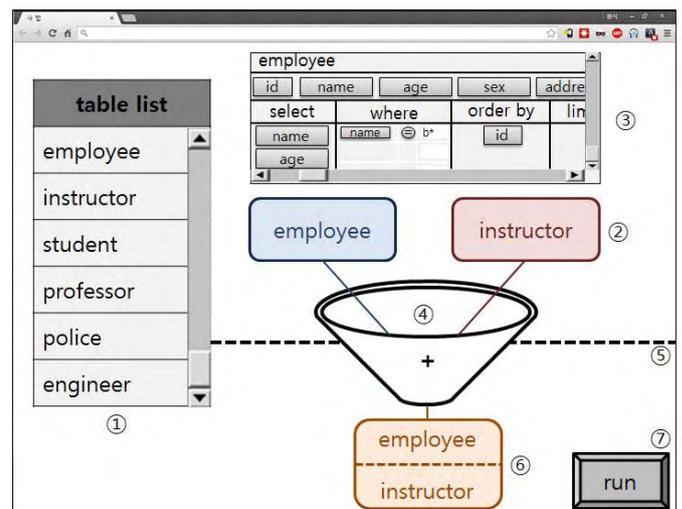
본 논문의 구성은 다음과 같다. 2장에서는 제안하는 인터페이스의 구성 요소 및 질의 처리 방법에 대해서 서술하고, 3장에서는 제안하는 인터페이스를 기반으로 다양한 질의를 작성하는 예제를 소개한다. 마지막으로 4장에서는

결론과 향후 연구에 대해 기술한다.

2. 인터페이스 구성 요소 및 질의 수행 방법

2.1 인터페이스 구성 요소

본 장에서는 제안하는 인터페이스의 전체 구성을 기술한다. (그림 1)은 제안하는 인터페이스를 나타낸다.



(그림 1) 제안하는 인터페이스 전체 구성

(1) 테이블 리스트

테이블 리스트는 현재 접속해있는 DB에서 사용자가 액세스할 수 있는 모든 테이블 목록을 보여준다. 테이블 리

*Corresponding author

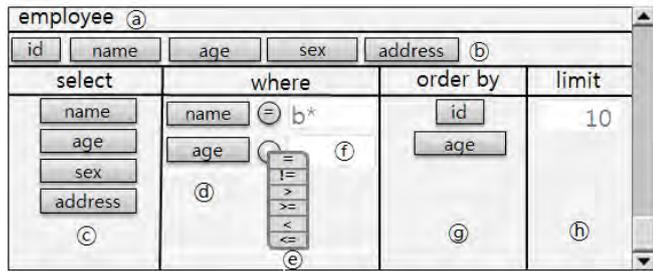
스트에서 선택하고자 하는 테이블을 테이블 리스트 밖으로 드래그 앤 드롭하면 단일 테이블 view 객체(2)가 생성된다.

(2) 단일 테이블 view 객체

단일 테이블 view 객체는 테이블 리스트에서 테이블을 테이블 리스트 밖으로 드래그 앤 드롭하여 생성할 수 있다. 바로 생성된 view 객체는 단일 테이블을 select할 수 있으며, view 객체를 클릭함으로써 자세한 설정을 적용하기 위한 view 설정 메뉴(3)에 접근할 수 있다. 다수의 테이블을 JOIN 또는 SUB-QUERYING하는 객체는 혼합기(4)를 통해 생성할 수 있으며, 이는 다중 테이블 view 객체(6)라고 불린다.

(3) view 설정 메뉴

view 설정 메뉴는 view 객체 각각의 설정을 변경한다. view 메뉴를 열면 해당 객체(a)가 액세스할 수 있는 컬럼이 UI의 상단(b)에 나열된다. 단일 테이블 view 객체일 경우 액세스할 수 있는 컬럼은 해당 테이블의 모든 컬럼이며, 다중 테이블 view 객체일 경우 액세스할 수 있는 컬럼은 해당 객체를 생산한 혼합기에서 select한 컬럼이다. 자세한 설정 메뉴는 (그림 2)와 같다.



(그림 2) view 설정 메뉴

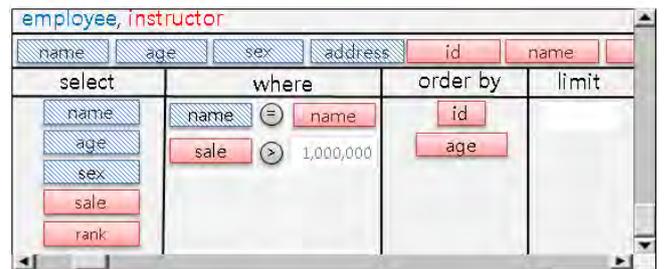
①타이틀은 설정을 적용할 객체의 이름이다. 객체의 형태에 따라 i)단일 테이블 view 객체일 경우 해당 테이블 이름이며, ii)다중 테이블 view 객체일 경우 액세스 가능한 모든 테이블 명이 '+' 연산자로 붙어있는 형태로 나타나며, iii)혼합기일 경우 액세스 가능한 모든 테이블 명이 ',' 기호로 연속되어있는 형태로 표기된다. ②컬럼리스트는 view 객체가 액세스할 수 있는 모든 컬럼을 보여준다. ③select 영역은 어떤 데이터에 액세스할 것인가를 결정한다. 컬럼리스트에서 컬럼을 드래그하여 select 영역에 드롭하면 select 영역에 추가되며, 빈칸으로 둘 경우 컬럼리스트의 모든 컬럼을 추가한 것으로 간주한다. where 영역은 컬럼(④), 연산자(⑤), 조건 값(⑥)으로 이루어지며 컬럼은 select 영역과 마찬가지로 컬럼리스트에서 드래그하여 where 영역에 드롭하면 된다. where 영역에 새로운 컬럼이 추가되면 연산자와 조건 값 부분이 활성화된다. 예를 들어, (그림 2)의 where 영역은 데이터 중 'name' 컬럼이

'b'로 시작하는 열만 액세스하도록 설정되어있고, 현재 컬럼 age에 대한 연산자를 선택하는 모습이다. 조건 값은 리터럴 값을 직접 입력해도 되고 컬럼리스트에서 드래그 앤드롭하여 컬럼으로 지정할 수 있다. ⑦order by 영역은 데이터의 액세스 순서를 지정하는 곳이며 컬럼리스트에서 컬럼을 드래그하여 드롭하면 된다. ⑧limit 영역은 액세스할 데이터 열의 최대값을 설정하는 곳이며 지정하지 않으면 무한대의 열까지 액세스한다.

(4) 혼합기

혼합기는 view 객체들이 액세스하는 데이터들을 토대로 새로운 view 객체를 생성하며, 실제로 JOIN과 SUB-QUERY를 구현하는 핵심적인 역할을 담당한다. (그림 3)은 employee 테이블에 액세스하는 단일 테이블 view 객체와 instructor 테이블에 액세스하는 단일 테이블 view 객체를 혼합할 때, 혼합기의 view 설정 메뉴를 나타낸다. 그림에서 사선 처리된 컬럼은 employee의 컬럼이며, 사선 처리 되지 않은 컬럼은 instructor의 컬럼이다. 각 컬럼은 혼합기 상측에 연결된 단일 테이블 view 객체가 select 설정한 컬럼들이다.

혼합기를 통해 생성된 view 객체는, 접근하는 테이블 수에 따라 단일 테이블 view 객체 및 다중 테이블 view 객체로 분류된다.



(그림 3) employee와 instructor 테이블을 액세스하기 위한 혼합기 설정의 예

(5) 기준선

기준선은 혼합기에 의해 생성되며, 기준선 위의 view 객체를 변경하면 기준선 아래의 객체를 재정의해야 한다. 만약 사용자가 하나의 view 객체를 변경하면, 해당 객체 아래의 기준선 밑의 모든 view 객체는 사라지고 다시 정의해야 한다.

(6) 다중 테이블 view 객체

다중 테이블 view 객체는 2개 이상의 단일 테이블 view 객체가 혼합기에 입력되어 생성되며, 액세스할 수 있는 데이터 컬럼은 혼합기에서 select한 컬럼이다. (그림 1-3)을 따라 생성된 employee와 instructor 테이블 데이터를 액세스하는 다중 테이블 view 객체의 초기 view 설정 메뉴는 (그림 4)와 같다.



(그림 4) employee와 instructor 테이블 데이터를 액세스하는 다중 테이블 view 객체 설정 메뉴

(6) 실행 버튼

객체를 모두 생성 및 설정한 후 실행 버튼을 누르면 실제 질의를 수행하게 된다. 객체와 설정으로부터 질의를 생성하는 방법은 2.2장의 질의 생성 방법을 따른다.

2.2 질의 생성 방법

본 절에서는 인터페이스에서 만든 객체와 설정을 토대로 실제 DB에서 수행될 질의 SQL문을 생성하는 방법을 서술한다. 이를 위해, 사용자가 선택한 객체와 설정을 파싱하여 질의 형태로 변경하고, 데이터베이스에 전송한다. 이 과정은 최상단의 view 객체부터 시작해 최하단까지 순서대로 이루어지며 파싱 규칙은 <표 1>의 3가지 규칙을 따른다.

<p>Rule 1 : meet Single Table View Object</p> <p>Displace to next SQL -</p> <pre>SELECT [column on select space] FROM [title] WHERE [condition on where space] ORDER BY [column on order by space] LIMIT [number on limit space]</pre>
<p>Rule 2 : meet Mixer</p> <p>Displace to next SQL -</p> <pre>SELECT T_i. [column on select space] FROM [original view] as T_i WHERE T_i. [condition on where space] ORDER BY T_i. [column on order by space] LIMIT [number on limit space]</pre>
<p>Rule 3 : meet Multiple Table View Object</p> <p>if Multiple Table View Object used all of column on select space from column list and none of condition on where space, then just use input SQL from Mixer, else displace to next SQL -</p> <pre>SELECT [column on select space] FROM [input SQL from Mixer] WHERE [condition on where space]</pre>

ORDER BY [column on order by space]
LIMIT [number on limit space]

<표 1> 파싱 규칙

3. 인터페이스를 이용한 질의 작성 예

본 장에서는 제안하는 인터페이스 이용한 다양한 질의 작성 예제를 통해 제안하는 인터페이스의 효용성과 우수성을 보인다. 각 질의 예제는 모든 객체 생성과 설정을 완료한 후 실행버튼을 누르기 바로 직전의 상태이며, 객체의 설정이 없을 경우 view 객체 설정 메뉴는 생략한다.

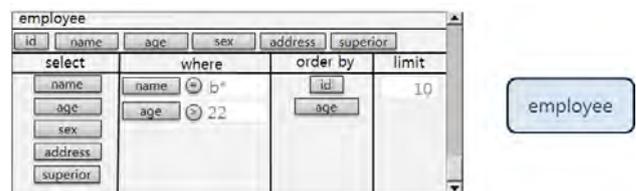
3.1. 단일 테이블 SELECT

<표 2>는 단일 테이블을 SELECT하는 질의문 예이다.

```
SELECT name, age, sex, address, superior
FROM employee
WHERE name='b*' AND age > 22
ORDER BY id, age
LIMIT 10
```

<표 2> 단일 테이블 SELECT 질의문 예

(그림 5)는 제안하는 인터페이스를 통해 <표 2> 질의문을 설정한 예이다.



(그림 5) 단일 테이블 select 예

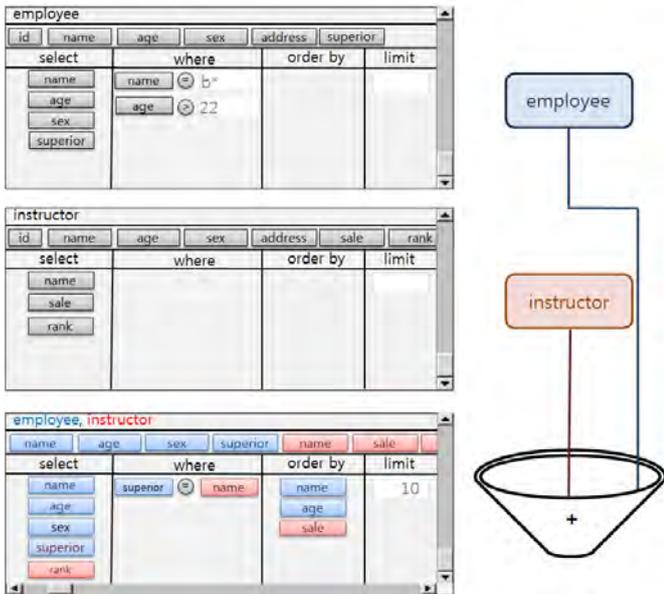
3.2. 2개 테이블의 JOIN

<표 3>은 2개의 테이블을 JOIN하는 질의문 예이다.

```
SELECT T1.name, T1.age, T1.sex, T1.superior
T2.rank, T2.sale
FROM employee as T1, instructor as T2
WHERE T1.name = 'b*' AND T1.age=22 AND
T1.superior = T2.name
ORDER BY T1.name, T1.age, T2.sale
LIMIT 10
```

<표 3> 2개 테이블의 JOIN 질의문 예

(그림 6)는 제안하는 인터페이스를 통해 <표 3> 질의문을 설정한 예이다.



(그림 6) 2개 테이블의 join

3.3. 3개 테이블의 JOIN 및 SUB-QUERYING

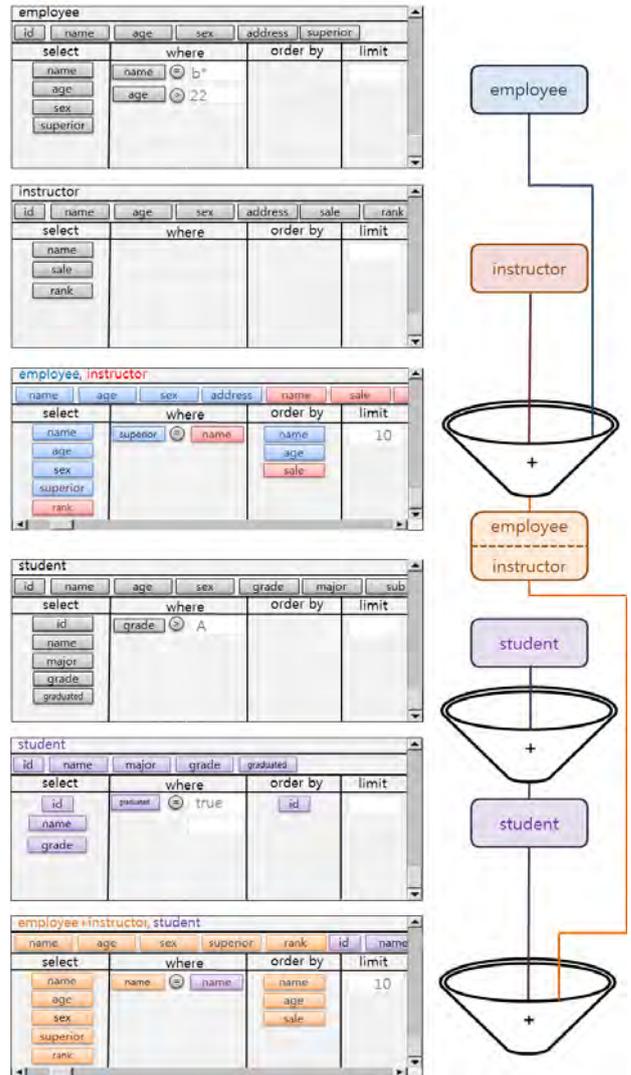
<표 4>는 3개의 테이블에 대해서 JOIN 및 SUB-QUERYING하는 질의문 예이다.

```

SELECT T1.name, T1.age, T1.sex, T1.superior
      T1.rank, T1.sale, T2.id, T2.grade
FROM   (SELECT T1.name T1.age, T1.sex,
              T1.superior, T2.rank, T2.sale
        FROM employee as T1, instructor as T2
        WHERE T1.name = 'b*' AND T1.age=22
              AND T1.superior = T2.name
        ORDER BY T1.name, T1.age, T2.sale
        LIMIT 10) T1,
        (SELECT id ,name, grade
        FROM (SELECT id, name, major,
                    grade, graduated
              FROM student
              WHERE grade > A)
        WEHRE graduated = true
        ORDER BY id) T2
WHERE T1.name = T2.name
ORDER BY T1.name, T1.age, T1.sale
LIMIT 10
    
```

<표 4>3개의 테이블에 대한 JOIN 및 SUB-QUERYING 질의문 예

(그림 7)은 제안하는 인터페이스를 통해 <표4>의 질의문을 설정한 예이다. employee와 instructor에 다중 테이블 view 객체에 대한 view 설정 메뉴는 아무 설정도 하지 않았기 때문에 생략했다



(그림 7) 3개 테이블의 join 및 sub-querying 예

4. 결론 및 향후 연구

본 논문에서는 일반 사용자를 위한 GUI 기반 DB 액세스 기법을 제안하였다. 이를 위해 select, join, sub-querying를 위한 인터페이스를 설계하였다. 향후 연구로는 본 논문에서 제안한 인터페이스를 구현하여 성능의 우수성을 검증하는 것이다.

참고문헌

- [1] Manyika, James, et al. "Big data: The next frontier for innovation, competition, and productivity." (2011).
- [2]Golany, Boaz, and J. E. Storbeck. "A data envelopment analysis of the operational efficiency of bank branches." Interfaces 29.3 (1999): 14-26.
- [3]Lohr, Steve. "The age of big data." New York Times 11 (2012).
- [4]Swan, Melanie. "The quantified self: Fundamental disruption in big data science and biological discovery." Big Data 1.2 (2013): 85-99.
- [5]Berman, Jules J. Principles of big data: preparing, sharing, and analyzing complex information. Newnes, 2013.
- [6]<http://www.navicat.com/> [7]<http://www.sequelpro.com/>
- [8]<http://www.heidisql.com/>