

표준 SQL 을 이용한 가계도 검색방법

도효진, 전부상, 변정용
동국대학교 컴퓨터공학과
byuniv@dongguk.ac.kr

Retrieving Family-Tree Using Standard SQL

Hyo-Jin Dho, Boo-Sang Jeon and Jeongyong Byun
Dept. of Computer Science, Dongguk University

요 약

인터넷 사용인구가 늘어나면서 다양한 종류의 정보를 제공하는 곳이 늘고 있다. 족보 사이트나 가계도 검색 사이트 같은 서비스도 개발되었다. 그 중에서 가문별 가계도 검색 서비스는 특정 가계나 특정 데이터베이스에 종속되는 경향이 있다. 이에 본 논문에서는 어떠한 환경에서도 원하는 가계도를 검색할 수 있도록 표준 SQL 질의문과 JAVA 를 사용하여 RDBMS 상의 가계 관계의 가계도를 표현하는 방법을 제안한다.

1. 서론

인물정보와 같은 실세계의 자료의 구조에서 계층형 자료 표현의 요구가 있다. 현재 관계형 데이터베이스 관리 시스템(DBMS)에서 이들을 표현하고 검색하는 방법이 필요하다. 특히 조선시대 인물정보 시스템(KoPIS)[6]은 가계도 정보 표현과 검색은 바로 이러한 요구의 대표적인 예이다. 또한 검색 결과는 웹에 그림으로 표현하여 서비스해야 하는 요구를 가지고 있으며, 오라클 DBMS[1]를 이용하여 시스템을 개발하였다. 일반적으로 가계도 정보는 오라클은 트리 검색을 위하여 SELECT 문에서 표준에는 없는 검색 조건에 별도의 편의 기능을 첨가하여 제공하고 있다. 하지만 검색 시스템이 DBMS 에 대하여 보다 보편성을 가지려면 표준 SQL 로 가계도 검색 질의를 구성해야 한다.

본 논문은 오라클 DBMS 를 이용하여 가능하였던 가계 트리 검색 질의어를 표준 SQL 로 대체하는 방법을 제안한다. 이를 위하여 실제로 구현된 가계도 스키마와 오라클의 SQL 사투리 표현법을 예시하고 이들에 대한 표준 SQL 표현 방법이 동치임을 보인다. 여기서 트리 검색은 관계대수로 표현이 어렵기 때문에 호스트 언어의 지원을 통하여 두 가지 질의의 결과를 통하여 검증한다.

2. 기존의 연구현황

오라클은 트리 구조 검색을 위하여 SELECT 문의 조건절에서 START WITH, CONNET BY PRIOR 와 같은

속성을 추가하여 트리 구조 검색을 편리하게 제공하고 있다. 하지만 이것은 오라클의 SQL 사투리에 해당하며 표준인 SQL-92 에는 이와 같은 기능이 추가되어 있지 못하다. 표준 SQL 로 표현하려고 할 때 SQL 로 직접 질의를 할 경우에는 검색이 불가능하며 ESQL 로 호스트 언어의 도움을 받아서 가능하다. C 와 같은 호스트 언어는 CURSOR 와 FETCH 문을 사용하고, 자바 언어의 경우 자바언어 자체에서 커서 기능을 내포하고 있기 때문에 getString 과 next 를 이용하여 쉽게 구현할 수 있도록 하고 있다.

3. 가계도 검색 시스템의 설계

3.1 요구 분석

가계도 검색 시스템을 다음과 같은 사항을 고려하여 설계한다.

첫째, 인물정보 데이터베이스는 RDBMS 를 사용한다. RDBMS 이외의 데이터베이스를 사용할 경우, 어떤 정보의 변경 시 그 정보와 관련된 모든 정보를 수정해주어야 하는 반면, RDBMS 를 사용하면 개개의 데이터를 일일이 관리하지 않더라도 개별적인 데이터를 연관성 있게 결합하여 통합적으로 관리를 할 수 있다.

둘째, 가계도 표현을 위해서 일반 가계도와 같은 트리 구조로 표현하기 위해 해당하는 데이터를 트리 형태로 검색해서 가져와야 한다.

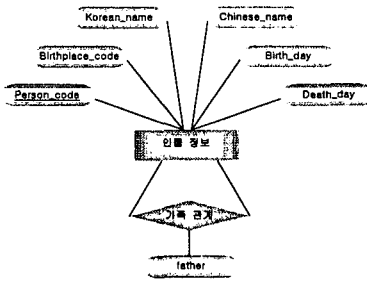
셋째, 가계도 검색 시스템은 특정 데이터베이스에

중속되지 않아야 한다. 그러기 위해선 특정 데이터베이스에 사용되는 질의문이 아닌 표준 SQL 문을 사용하여 데이터를 검색하고 질의해야 한다.

3.2 개체 관계 다이어그램

데이터베이스의 각 인물의 정보는 개인 정보와 가족 관계가 있다. 개인 정보에는 기본 키로 각 인물의 고유 번호인 인물 코드와 본관을 나타내는 본관 코드 그리고 한글 이름, 한자 이름, 출생일자, 사망일자가 있다. 가족 관계에는 부자 관계가 있다.

이것을 RDBMS의 개체-관계 모델을 이용해 나타내면 그림 1과 같은 개인 정보에 대한 개체-관계 다이어그램을 얻을 수 있다.



[그림 1] 개인 정보에 대한 개체-관계 다이어그램

3.3 스키마 설계

앞 절의 개체-관계 다이어그램 설계를 이용하여 관계형 데이터베이스에서 스키마 설계를 기술한다. 타입은 character 형태이다.

[표 1] 인물 정보

필드	인물 코드	본관 코드	한글 성명	한자 성명	출생 일자	사망 일자	아버지
	Person_code	BirthPlace_code	Korean_name	Chinese_name	Birth_day	Death_day	Father
크기	8	5	8	8	8	8	8

인물 정보는 한 인물의 개별사항을 나타내고 있다. 인물 코드는 인물 고유의 번호를 가리키고, 본관 코드는 인물의 본관을 본관 목록에서 검색한 본관 코드를 가리킨다. 한글 성명, 한자 성명, 출생일자, 사망일자는 각 인물의 성명 및 생몰 일자를 가리킨다. 이 테이블에서 본관코드는 외래키로서 본관 목록에서 가져와서 사용한다. 아버지 필드는 인물 코드로 되어있고 가족 관계인 부자관계를 표현하는데 사용한다.

3.3 검색 질의 처리

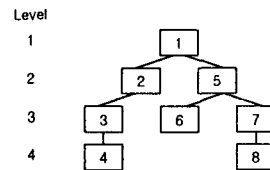
특정 가계에 대한 가계도를 검색하기 위해서는 최대 선조 검색 방법과 후손 검색 방법을 사용하게 되는데, 먼저 특정 가계에 대한 최대 선조를 검색하고 검색된

최대 선조에 대한 모든 후손을 검색한다. 모든 후손이 검색되어지면 특정 가계에 속해 있는 모든 인물이 검색되어진 것이다.

최대 선조 검색 방법은 특정 가계에 대한 최대 선조를 검색하는 것으로 인물 정보 데이터베이스의 부자 관계를 이용하여 검색한다. 검색 방법은 먼저 입력되어진 특정 인물에 대한 아버지를 검색하고 다시 검색된 아버지에 대한 아버지를 검색한다. 이러한 과정을 최대 선조가 검색되어질 때까지 계속한다. 최대 선조는 아버지에 대한 정보가 없는 인물을 최대 선조로 간주한다. 아버지에 대한 정보가 없는 인물이 검색되어지면 그 인물이 최대 선조가 된다.

후손 검색 방법은 특정 가계의 최대 선조에서 인물 정보 데이터베이스의 부자 관계를 이용하여 후손을 검색한다. 검색 방법은 선택된 최대 선조를 아버지 필드로 가지는 인물들을 검색한다. 검색된 인물들을 가지고 다시 검색된 인물을 아버지 필드로 가지는 인물들을 검색한다. 이런 식으로 검색을 하다가 해당 인물이 안 나오면 윗대에서 검색된 인물들이 여러 명일 경우, 각각에 대해서 다시 위와 같은 과정을 반복한다. 최종적으로 모든 인물들에 대해서 자식을 가지지 않을 경우까지 검색한다.

후손 검색 방법에서 트리 구조로 데이터를 표현하기 위해 먼저 최대 선조의 자식에 대해서 자식이 여러 명일 경우 첫번째 자식에 대해서 먼저 후손을 찾고 나서 두 번째, 세 번째 자식에 대해 후손을 찾는다. 이렇게 처음 검색한 첫번째 자식에 대해 검색이 다 끝나면 두 번째 자식에 대해서도 후손을 찾고, 두 번째 자식의 후손에서도 첫번째 자식을 우선으로 해서 검색한다. 즉, PRE ORDER 방식으로 후손들을 검색하고 각 레벨을 정하게 되면 그림 2와 같은 트리 형식으로 나타낼 수 있다.



[그림 2] 인물 검색 순서

4. 구현

가계도를 검색하기 위한 요구사항을 만족시키기 위해 특정 데이터베이스에 중속되지 않고 SQL 문을 사용하기 위해, 자바로서 데이터베이스에 접근하여 가계도를 검색하는 모듈을 구현했다.

자바를 사용하면 소스 코드의 큰 변화 없이 쉽게 다양한 플랫폼에서 다양한 데이터베이스에 적용시킬 수 있다.

그리고 각 데이터베이스마다 지원하는 SQL 문이 달라질 수 있기 때문에 가계도 검색 시 필요한 트리 검

색을 위해 표준 SQL 문을 사용했다. 이는 특정 RDMS 에서 SQL 확장의 한 형태로 트리 검색을 SQL 문으로서 지원하기 때문에 특정 데이터베이스에 구애 받지 않기 위해서 표준 SQL 문만을 사용했다. 실제로 ORACLE 에서는 'start with'절과 'connect by prior' 절을 이용하여, 한 문장만으로 트리 검색이 가능한 질의문을 만들 수 있지만, 다른 데이터베이스에서는 그 구문이 존재하지 않기 때문에 E-SQL 형식으로 자바와 함께 질의문을 구성하여 트리 검색을 구현했다.

ORACLE 에서의 계층구조를 표현할 때 시작위치가 되는 노드나 계층을 나타내는 'start with'절과 부모 노드와 자식 노드 사이의 관계를 구분짓는 'connect by prior'절에 대응하는 가계도 검색을 위해, 선택된 인물의 최대 선조를 먼저 찾은 최대 선조 검색 모듈과 최대 선조로부터 시작되는 모든 후손들을 검색하는 후손 검색 모듈을 구현했다.

4.1 최대 선조 검색 모듈의 구현

최대 선조의 검색은 특정 인물에 대한 최대 선조를 검색하는 것으로 검색 방법은 먼저 특정 인물에 대한 아버지를 검색하고 다시 검색된 인물에 대한 아버지를 검색한다. 이러한 과정을 아버지에 대한 정보가 없는 인물이 검색될 때까지 계속 검색해 간다. 아버지가 없는 인물이 검색되면 그 인물의 특정 가계의 최대 선조가 된다.

먼저 오라클에서 제공되는 'start with'절과 'connect by prior' 절을 사용하여 최대 선조 검색을 한 방법이다. 'Person_code'의 Father 필드를 가지는 아버지의 인물코드(person_code)를 검색하고, 다시 그 아버지의 아버지 인물코드를 검색하여 최대 선조를 검색한다.

```
Select person_code From new_person_info
Where father = null
Start with person_code = 'person_code'
Connect by prior father='person_code';
```

[그림 3] 최대 선조 검색 모듈 - 오라클

표준 SQL 문에서는 루프를 돌면서 아버지 정보를 계속해서 찾으며 최대 선조를 검색한다.

```
while(chk){
String father_query ="select father
from new_person_info where person_code
='"+code+"'";
result.next();
if(result.getString(1) == null)
chk = false;
else code = result.getString(1);
}
```

[그림 4] 최대 선조 검색 모듈 - 표준 SQL

후손 검색은 검색된 최대 선조에서 후손들을 검색해 나가는데, 먼저 선조를 아버지 필드로 하는 인물들을 검색한다.

오라클에서는 'person_code'를 아버지 필드로 가지는 인물코드(person_code)들을 검색하여 후손을 검색하게 된다.

```
Select person_code From new_person_info
Start with person_code = 'person_code'
Connect by prior 'person_code' = father;
```

[그림 5] 후손 검색 모듈 - 오라클

일반 SQL에서는 같은 아버지를 둔 형제들이 나타날 수 있으므로 while 문에서 검색된 인물들을 배열에 저장한다.

```
String query = "select person_code from
new_person_info where father = " + code;
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(query);
level++;
while(rs.next()){
arr[cnt][0] = rs.getString(1);
//검색된 인물코드
arr[cnt][1] = cnt+""; //인덱스
arr[cnt][2] = level+""; // 레벨값
cnt++;
.....
}
```

[그림 6] 후손 검색 모듈 1- 일반 SQL

후손 검색에서 자식이 없는 경우는 선대의 아버지대에서 아버지의 형제들이 있는 경우 아버지의 형제들에서 다시 후손 검색을 한다. 즉, recursive call 형식으로 후손 검색시 후손이 없을 경우, 질의문이 null 인 경우 선대로 올라가서 후손을 검색하는 방법을 취했다.

트리 형식으로 나타내기 위해 후손의 좌측 자식부터 검색하기 때문에 PRE ORDER 방식이 된다.

```
public String child_relat_search(int leve,String
code)
{
.....
for(;;){
if(code == null){ // 해당인물이 없다
break;
}
.....
isnull =
child_relat_search(level,isnull);
// 재귀호출
if(isnull == null){ // 자식이 없을 경우
code = null;
break;
}
.....
}
```

[그림 7] 후손 검색 모듈 2- 일반 SQL

5. 결론

본 실험은 가계 정보를 제공하기 위한 방법인 가계도의 표현이 오라클의 사투리 SQL 를 모든 데이터베이스에서 가능하도록 하기 위해 표준 SQL 을 사용하여 가계도를 표현하는 방법을 제안하고, 또한 웹 기반 서비스 요구를 위하여 특정 플랫폼과 데이터베이스에서 구애 받지 않고 사용할 수 있는 언어인 JAVA 를 ESQL 의 호스트 언어로 해서 가계 정보의 트리 검색 방법을 구현하여 그들의 결과가 같음을 보았다.

가계 정보를 검색하는 방법으로는 최대 선조 검색 방법과 후손 검색 방법을 사용했는데, 최대 선조 검색 방법과 선조에 대한 후손 검색 방법은 모든 데이터베이스에서 표준 SQL 로 검색이 되었고 데이터베이스에 정보가 저장되어 있는 인물에 대해서 가계 정보 데이터의 양에 관계없이 가계도의 표현이 가능했다. 이 결과는 오라클의 검색문의 결과와 같았다. 결국 표준 SQL 를 채택하고 있는 DBMS 에서 이 시스템을 장착할 수 있게 된 것이다. 하지만 가계도 검색의 속도 면에 있어서는 오라클의 트리 구조 검색 사투리를 쓰는 것이 표준 SQL 과 호스트 언어를 결합한 방법 보다 훨씬 빠른 것으로 사려된다.

참고문헌

- [1] Cooperative Server Technology, Oracle7 Server SQL Reference Release 7.2, ORACLE
- [2] P.Chan, R.Lee, D.Kramer, "The Java Class Libraries Second Edition, Vol 1 supplement for the java 2 platform standard Edition, v1.2
- [3] Krishina Sankar, "Java 1.2 Class Libraries", SAMS
- [4] 국조문과방목(國朝文科榜目), 태학사.
- [5] 엄채임, 웹기반 조선시대 인물 정보 서비스 시스템의 설계 및 구현, 동국대학교대학원 전자계산학과 석사학위논문
- [6] 정보통신부, 웹에서 클라이언트/서버를 이용한 조선시대 역사인물 정보검색 시스템 개발에 관한 연구:1 차년도, p.265, 1999
- [7] 이창걸, 조선중기 지배엘리트의 충원에 관한 실증적 연구, 고려대학교대학원사회학과 박사학위논문
- [8] 디지털 한국학 " [http:// www.koreandb.net](http://www.koreandb.net)"
- [9] 경주 김문계림군과 <http://web.eyes.co.kr/~djh>