김일성종합대학학보
(자연과학)
주체104(2015)년 제61권 제7호

JOURNAL OF **KIM IL SUNG** UNIVERSITY
(NATURAL SCIENCE)
Vol. 61 No. 7 JUCHE104(2015).

SQL질문해석에 의한 MySQL, PostgreSQL의 View변환의 한가지 방법

리충혁, 윤희광

SQL(Structured Query Language)은 자료기지로부터 정보를 생성, 복구, 조종하기 위한 표준질문언어이다.[1]

View는 SQL표준에서 정의한 보조기구로서 자료기지에 보관된 정보들을 효률적으로 탐색하여 리용할수 있도록 하며 자료기지에 보관된 자료들을 임의의 형식으로 보관된 가상표처럼 리용할수 있도록 조직해주는 기능을 수행한다.[2]

View는 Create View에 의하여 생성되는데 SQL표준에서 정의한 Create View의 형식은 다음과 같다.[1]

Create View <View이름> [<View 마당목록>] AS <질문표현식> [With CHECK OPTION]

MySQL과 PostgreSQL은 모두 관계형자료기지관리체계들로서 SQL표준을 따르고있지만 고유한 설계방식과 높은 성능을 지향하는것으로 하여 SQL표준의 기본적인 부분만 구현하고 대부분의 질문형식들을 확장변경하여 구현하고있다. 그런것으로 하여 Create View질문에서 <질문표현식>과 [With CHECK OPTION]부분들은 MySQL과 PostgreSQL에서 많은 차이점을 가지고있다. 이러한 차이점으로 하여 이미 개발된 자료기지변환프로그람들에서는 MySQL과 PostgreSQL사이의 View변환을 사용자가 수동작업으로 진행하도록 하고있다.

MySQL과 PostgreSQL에서 리용하고있는 View정의문들을 비교한 결과는 표 1과 같다.

표 1. MySQL과 PostgreSQL의 Create View명령문 비교결과

MySQL	PostgreSQL
Create [or Replace]	Create [or Replace]
[Algorithm={Undefined | Merge | Temptable}]	[Temp | Temporary]
[Definer={ USER | CURRENT_USER }]	
[SQL SECURITY { DEFINER | INVOKER }]	
View 이름 [(마당목록)]	View 이름 [마당목록]
AS SELECT질문	AS 질문
[With [Cascaded | Local] Check Option	

이상의 선행연구에 기초하여 론문에서는 MySQL과 PostgreSQL의 View를 자동변환하기 위한 한가지 방법으로서 View정의문들을 구문해석하기 위한 구문해석기를 작성하고 구문해석된 View정의문들을 리용하여 MySQL과 PostgreSQL사이의 View변환을 위한 변환알고리듬을 작성하였다.

1. SQL구문해석에 의한 MySQL, PostgreSQL의 View변환방법

1) View정의문의 구문해석을 위한 구문해석기의 작성

구문해석기를 작성하기 위하여 현재 콤파일러작성도구로 널리 리용되고있는 Lex와 Yacc를 리용하여 MySQL과 PostgreSQL의 View정의문을 해석할수 있는 구문해석기를 각각 작성하였다.

MySQL의 View정의문을 구문해석하기 위한 첫 단계인 어휘해석을 위한 BNF문법은 다음과 같다.

```
stmt:
    Create v_options View v_name AS sel_stmt vw_options ';' END
```

문법에서 v_options는 View정의문에서 View예약어앞에 놓이는 선택항목들의 모임으로서 다음과 같다.

```
v_options:
    /*empty*/
    | v_option_list
    ;
v_option_list:
    v_option
    | v_option_list v_option
    ;
```

그리고 v_name은 View의 이름을 나타내며 sel_stmt는 select질문을 나타낸다.

```
sel_stmt:
    SELECT sp_options s_stmt sa_options upda_opt
    ;
```

또한 vw_options는 select문에서 From절뒤에 놓이게 되는 항목들을 나타낸다.

입력으로 받아들인 View정의문을 해석할 때 다음과 같은 항목들에 대해 식별값을 할당하여 보관하도록 한다.(표 2)

표 2. Create View명령문의 구성부분에 대한 식별값

식별값	해석부분	식별값	해석부분
0	Create View	5	연산자
1	View이름	6	함수이름
2	Select질문의 시작부분	7	함수파라메터렬
3	Select질문의 끝부분	8	기타 부분
4	상수값		

한편 PostgreSQL에 View정의문을 해석할수 있는 구문해석기도 PostgreSQL의 문법에 맞추어 MySQL에서와 같은 방식으로 작성한다. 마지막으로 MySQL과 PostgreSQL에서 리용하고있는 내장함수들의 목록을 작성하여 대응표를 작성한다. 이 내장함수대응표는 함수이름을 변환할 때 리용된다.

2) MySQL과 PostgreSQL사이의 View변환알고리듬

변환에서는 SQL구문해석에서 얻어진 부분들과 색인값들을 리용한다.

변환은 크게 두가지 단계로 이루어진다.

첫번째 단계는 구문해석에서 얻어진 값들을 변환하는 공정이다.

두번째 단계는 변환된 값들을 리용하여 목적자료기지체계에서 실행되는 View정의문을 생성하는것이다.

구문해석기에 의하여 값변환은 다음의 세가지 부류로 나누어 진행한다.

① 변환을 필요로 하지 않는 값은 그대로 넘긴다.

이런 값들로는 Create view 와 View이름, AS, SELECT, 연산자들이 속하는데 이것은 표 2의 식별값 0, 1, 2, 3에 대응된다.

② 변환해야 할 부분들에 대한 치환변환을 진행한다.

이런 값들로는 View정의문에서 호출하고있는 내장함수들의 이름이 속하는데 이것은 표 2의 식별값 6에 대응된다.

③ 변환불가능한 부분들은 삭제한다.

이러한 값들은 어느 한 자료기지체계에서만 특정하게 구현하고있는 부분들로서 표 2의 식별값 8에 대응되는 부분들이다.

치환변환이 끝나면 구문해석에서 얻어진 순서대로 다시 문자렬을 조합하여 Create view 정의문을 생성해낸다.

View변환의 전체적인 처리흐름도는 그림과 같다.

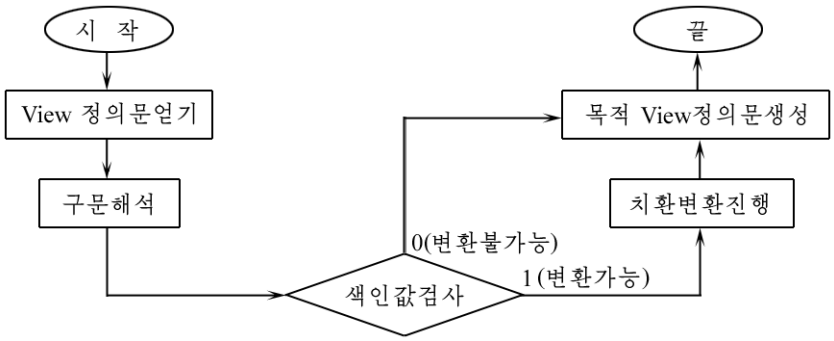

그림. View변환의 전체적인 처리흐름도

2. 성 능 평 가

론문에서 제안한 방법을 리용한 변환프로그람을 작성하여 MySQL에서 PostgreSQL에로 View변환시험을 진행하였다. 성능평가는 MySQL봉사기에서 서로 다른 형태의 View를 창조하고 그것들에 대한 자동변환을 진행하는 방법으로 진행하였다. 성능평가결과는 표 3과 같다.

표 3으로부터 고급한 SQL질문을 리용한 View들에 대하여서는 자동변환을 진행하지 못하였다는것을 알수 있다.

표 3. 성능평가결과

No.	View종류	자동변환성공(0)/실패(×)
1	간단한 Select질문만으로 구성된 View	○
2	간단한 문자렬처리 내장함수들을 리용한 Select질문으로 구성된 View	○
3	간단한 수학 및 통계함수를 리용한 Select질문으로 구성된 View	○
4	고급한 문자렬처리 및 수학, 통계함수들을 리용한 Select질문으로 구성된 View	×
5	내포질문을 리용한 View	×

맺 는 말

MySQL과 PostgreSQL의 View를 호상변환하기 위하여 Create View질문을 해석할수 있는 구문해석기를 작성하고 그것에 기초한 자동변환알고리듬을 작성하였다.

참 고 문 헌

[1] Clay Andres et al.; Beginning SQL Queries, Novice Professional, 264, 2008.

[2] Alan Beaulieu; Learning SQL, O'Reilly, 408, 2005.

주체104(2015)년 3월 5일 원고접수

A Method of Migrating View between MySQL and PostgreSQL using SQL Query Analysis

Ri Chung Hyok, Yun Hui Gwang

We proposed and implemented SQL query analysis method to migrate views automatically between MySQL and PostgreSQL, the most popular open source database system.

Key words: view, SQL, query analysis