Tạp chí Tin học và Điều khiển học, T.14, S.4 (1998) (73-80)

SQL QUERY EXPRESSED IN RELATIONAL CALCULUS

HO THUAN, TRAN THIEN THANH

Abstract. In this paper, we present some results on the equivalence between SQL queries and relation calculus expressions. These results are used to translate SQL queries into equivalent expressions of relational algebra.

I. INTRODUCTION

Stefano Ceri and Georg Gottlob, in [1] presented a translator from a subset of SQL queries into relational algebra. We have extended the results in [1] for subqueries with GROUP BY clause which can be nested at any level of a SQL query, and some extensions to condition of HAVING clause. This work is a basis step for this translation. In this paper, we have proved some results on the equivalence between SQL query and relational calculus expression.

II. BASIS KNOWLEDGE

1. Relational calculus

A relational calculus expression is of the form

$$\{t(\langle components \rangle) | \psi(t)\},$$

where

- t is a tuple variable;
- *components* is a list of components of the form:
  + A_i - is an attribute,
  + $R.A_i$ - R is a relation name, A_i is an attribute of R;
- $\psi(t)$ is a formula building from the atoms and collection of logical operators.

In order to use aggregate function in relational calculus, we extended the *components* to accept the form $F_i[A_j]$, where F_i is a function and A_j is an attribute, and other extensions to the atoms of formula ψ of some types such that: $F_i[A_j](s)\,\Theta\, u[A_i]$, $F_i[A_j](s)\,\Theta\, a$, where $F_i[A_j](s)$ is value of function F_i computed on attribute A_j for the tuple s, a is a constant, and $\Theta \in \{=, \neq, >, \geq, <, \leq\}$.

2. Structured Query Language (SQL)

a. Syntax of SQL query

SELECT ⟨*selector*⟩|* FROM ⟨*relation_list*⟩ [WHERE ⟨*predicate*⟩]
[GROUP BY ⟨*gb_attr*⟩ [HAVING ⟨*hav_condition*⟩]]

b. The meaning of clauses

The SELECT clause indicates attributes and functions are selected. The asterisk denote for all attributes of ⟨*relation_list*⟩.

The FROM clause indicates relations used for query.

Note: Every SQL query must have at least the SELECT clause and the FROM clause.

The WHERE clause indicates condition used to select tuples, only select tuples that satisfying the condition.

The GROUP BY clause indicates attributes, those used to group the tuples.

The HAVING clause indicates condition used to select groups, only select groups satisfying the condition.

c. The operators

The operators used to combining results of SQL queries: INTERSECT, UNION, MINUS.

3. Notation and relations used to illustrate

□ *Notation*

- *car(list of relational expressions)* indicates the Cartesian product of all the relational expressions.
- *attr(list of relations)* is the set of the attributes in the attributes schema of the specified relations.
- *attr(relational expressions)* is the set of attributes occurring in the results produced by the evaluation of a relational expression.
- *rels(list of attributes)* is the set of relations having the specified attributes.
- *rels(relational expression)* is the set of the relations, whose attributes appear in the relational expression.
- *extrattr(predicate)* is the set of attributes which appear in the *predicate*.
- *extrels(predicate)* is the set of relations whose attributes appear in the *predicate*.
- *meaning(Q)* is the relation results of query *Q*.

□ *The relations used to illustrate*

To illustrate, we use the database relations in Date's book [2].

- *Relations S - Suppliers*

SCODE	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris

- *Relation P - Products*

PCODE	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	Lodon
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London

- *Relation SP - Supplier-Product*

SCODE	PCODE	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S2	P1	300
S2	P2	400
S3	P2	400

III. SQL QUERIES EXPRESSED IN RELATIONAL CALCULUS

1. Notation and definition

Let Q be a SQL query of the form:

SELECT ⟨*selector*⟩ FROM ⟨*relarion_list*⟩ WHERE ⟨*predicate*⟩
GROUP BY ⟨*gb_attr*⟩ HAVING ⟨*hav_condition*⟩

- $Q.\langle component\rangle$ denotes the corresponding components of Q,
- $Q.\langle component\_1\rangle.\langle component\_2\rangle$ denotes the corresponding component of $Q.\langle component\_1\rangle$,
- $Q.\langle selector\rangle.\langle attrs\rangle$ is list of attributes in $Q.\langle selector\rangle$,
- $Q.\langle selector\rangle.\langle function\_list\rangle$ is list of functions in $Q.\langle selector\rangle$.

In this paper we assume $Q.\langle selector\rangle.\langle function\_list\rangle = \{F_i[A_j]\}$, $i = 1, ..., k$, $F_i[A_j]$ is a function F_i computed on attribute A_j.

Example: Let R be relation, R has the schema $R(A, B)$. Q be a SQL query of the form SELECT $R.A$, $R.B$, SUM($R.A$) FROM R, then

$Q.\langle selector\rangle = \{R.A,\ R.B,\ \mathrm{SUM}(R.A)\}$
$Q.\langle selector\rangle.\langle attrs\rangle = \{R.A,\ R.B\}$
$Q.\langle selector\rangle.\langle function\_list\rangle = \{\mathrm{SUM}(R.A)\}$

- $F_i[R; U; A_j](t)$ (where R is a relation, U is a subset of $attr(R)$, A_j is a attribute of R, $A_j \notin U$, t is a tuple in R): is the value of function F_i computed on attribute A_j of relation R with group-attribute U for tuple t of R.

Note: The values of $F_i[A_j]$ are same with tuples in a group (Fig. 1).
$\forall t, t' \in R$, if $t'(U) = t(U)$ then $F_i[R; U; A_j](t') = F_i[R; U; A_j](t)$

...	U	$F_i[R; U; A_j]$
r_1	g_1	f_1
r_2	g_1	f_1
r_3	g_2	f_2
r_4	g_2	f_2
r_5	g_2	f_2

Fig. 1

Definition 1. Let Q be a SQL query and E be a relational calculus expression, we say that Q is equivalent to E iff the results of Q and E are the same when we substitute the same relations for identical name in the two expressions.
When Q is equivalent to E, we say that E is Q expressed in relational calculus and denoted $Q = E$.

Definition 2. Let Q, Q' be SQL queries. We say that Q is equivalent to Q', denoted $Q = Q'$, iff when we substitute the same relations for identical name in the two expressions, we get the same result.
We have $Q = Q' \Leftrightarrow meaning(Q) = meaning(Q')$.

2. The top level query

Let Q be a SQL query of the form:

SELECT ⟨*selector*⟩ FROM ⟨*relation_list*⟩ WHERE ⟨*predicate*⟩
GROUP BY ⟨*gb_attr*⟩ HAVING ⟨*hav_condition*⟩

Based on the meaning of Q, we have:

$$Q = \{t(Q.\langle selector\rangle) \mid \exists t'(R(t') \wedge t(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle) \wedge F(t') \wedge \forall r(R(r) \wedge (r(Q.\langle gb\_attr\rangle) = t'(Q.\langle gb\_attr\rangle) \rightarrow H(r))) \wedge (t(F_i[A_j]) = F_i[R''; Q.\langle gb\_attr\rangle; A_j](t')\ i = 1, ..., k)\} \quad (1)$$

where

$R = car(Q.\langle relation\_list\rangle)$,
$F(t)$ is predicate $Q.\langle predicate\rangle$,
$H(t)$ is condition $Q.\langle hav\_condition\rangle$,
$R' = \{t \mid R(t) \wedge F(t)\}$ - set of tuples of R, those are satisfying $Q.\langle predicate\rangle$,
$R'' = \{t \mid R'(t) \wedge \forall t'(R'(t') \wedge (t'(Q.\langle gb\_attr\rangle) = t(Q.\langle gb\_attr\rangle) \rightarrow H(t')))\}$.

We rewrite equation (1):

$$Q = \{t(Q.\langle selector\rangle) \mid \exists t'(R''(t') \wedge t(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle) \wedge (t(F_i[A_j]) = F_i[R''; Q.\langle gb\_attr\rangle; A_j](t')\, i = 1, ..., k)\} \quad (2)$$

3. The subquery

The subqueries may be used in conjunction with the IN, ALL, EXISTS, ... operators.

Example. *Find the name of suppliers, those do not supply product P1.*

Q= SELECT SNAME FROM S WHERE "P1" NOT IN
(SELECT PCODE FROM SP WHERE SCODE=S.SCODE)

We consider the subquery:

Q' = SELECT PCODE FROM SP WHERE SCODE=S.SCODE

Remark.

1) The subquery Q' is dependent on S.SCODE of S.
2) To have *meaning* (Q), we need to have *meaning* (Q').

Definition 3 (External relation, External attribute of subquery).
Let Q be a subquery, R be a relation, if there exists attribute(s) of R appear in Q, but R not in $Q.\langle relation\_list\rangle$ then R is called external relation of Q.

Every attribute of Q's external relation is called external attribute of Q.

The set of all external attributes of Q is denoted by $Other(Q)$.

Definition 4 (The relation result of subquery).
Let Q be a subquery, $Other(Q)$ is set of all external attributes of Q. $S = car(rels(other(Q)))$.
For each $s \in S$, $Q(s)$ is a subquery, it is obtained by replace each attribute $A_i \in Other(Q)$ by $s(A_i)$.
Let $R_s = meaning(Q(s))$.

S

$Other(Q)$
s_1
s_2
s_3

R_{s_1}

$Q.\langle selector\rangle$
r_1
r_2
r_3

R_{s_2}

$Q.\langle selector\rangle$
r_4
r_5

R_{s_3}

$Q.\langle selector\rangle$

$R_{s_3} = \emptyset$

R

$Q.\langle selector\rangle$	$Other(Q)$
r_1	s_1
r_2	s_1
r_3	s_1
r_4	s_2
r_5	s_2

Fig. 2. Relation result of subquery

Definition: The relation result of subquery Q is defined by expression:

$$\{(r, s) \mid S(s) \wedge R_s(r)\} \quad (3)$$

Example: Let Q be a subquery

SELECT PCODE FROM SP WHERE SCODE=S.SCODE

$Other(Q)$ = {SCODE, SNAME, STATUS, CITY}

$s_1 = (S1,$ Smith,20,London)
$s_2 = (S2,$ Lones,10,Paris)
$s_3 = (S3,$ Blake,30,Paris)

R_{s_1}

PCODE
P1
P2
P3

R_{s_2}

PCODE
P1
P2

R_{s_3}

PCODE
P2

The relation result of subquery Q

R

PCODE	S.SCODE	SNAME	STATUS	CITY
P1	S1	Smith	20	London
P2	S1	Smith	20	London
P3	S1	Smith	20	London
P1	S2	Jones	10	Paris
P2	S2	Jones	10	Paris
P2	S3	Blake	30	Paris

By equation (1), we have:

$$R_s = \{t(Q.\langle selector\rangle) \mid \exists t'(R_s^*(t') \wedge t(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle) \wedge (t(F_i[A_j]) = F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t')\, i = 1, ..., k))\},$$

where

$$R_s^* = \{r \mid R(r) \wedge F(r, s) \wedge \forall r'(R(r') \wedge F(r', s) \wedge (r(Q.\langle gb\_attr\rangle) = r'(Q.\langle gb\_attr\rangle) \rightarrow H(r', s)))\}$$

IV. RESULTS

Theorem 1. *Let Q be a subquery of the form*
SELECT $\langle selector\rangle$ FROM $\langle relation\_list\rangle$ WHERE $\langle predicate\rangle$ GROUP BY $\langle gb\_attr\rangle$
HAVING $\langle hav\_condition\rangle$
then

$$Q = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists(P_{FH}(p) \wedge t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = p(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) \wedge (t(F_i[A_j]) = F_i[P_{FH}; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p)\, i = 1, ..., k))\}$$

where

$P = car(Q.\langle relation\_list\rangle \cup rels(Other(Q))),$
$P_F = \{p \mid P(p) \wedge F(p)\},$
$P_{FH} =$
$\{p \mid P_F(p) \wedge \forall p'(P_F(p') \wedge p'(Q.\langle gb\_attr\rangle \cup Other(Q)) = p(Q.\langle gb\_attr\rangle \cup Other(Q)) \rightarrow H(p'))\}.$

Define $P^* = \{(t, s) \mid S(s) \wedge R_s^*(t)\}$. The proof of Theorem 1 is based on Lemma 1, and Lemma 2. We omit here the proof of these lemmas.

Lemma 1. *We have* $P^* = P_{FH}$.

Lemma 2. *For every* $p \in P^*$, $p = (t, s)$, *where* $s \in S$, $t \in R_s^*$. *Let* F_i *be a function,* $A_j \in attr(Q.\langle relation\_list\rangle)$ *we have:*

$$F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t) = F_i[P^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p) \quad (*)$$

Proof of Theorem 1. By equation (3) we have the equivalence of Q with the expression E defined by:

$$E = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists s\,(S(s) \wedge t(Other(Q)) = s \wedge \exists r\,(R_s(r) \wedge t(Q.\langle selector\rangle) = r))\}$$
$$= \{(r, s) \mid S(s) \wedge R_s(r)\},$$

where

$$R_s = \{t(Q.\langle selector\rangle) \mid \exists t'(R_s^*(t') \wedge t(Q.\langle selector\rangle.\langle attrs\rangle)) = t'(Q.\langle selector\rangle.\langle attrs\rangle) \wedge (t(F_i[A_j] = F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t')\, i = 1, ..., k))\}.$$

Let

$$E' = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists p(P_{FH}(p) \wedge t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = p(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) \wedge (t(F_i[A_j]) = F_i[P_{FH}; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p)\, i = 1, ..., k))\}.$$

By Lemma 1, we have

$$E' = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists p(P^*(p) \wedge t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = p(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) \wedge (t(F_i[A_j]) = F_i[P^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p)\, i = 1, ..., k))\}.$$

We show that $E = E'$

a) First we show that $E' \subseteq E$

$\forall t \in E' \Rightarrow \exists p \in P^*$.

$t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = p(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q))$

$t(F_i[A_j]) = F_i[P^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p)\, i = 1, ..., k$.

Since $p \in P^*$ then $p = (r, s)$ where $s \in S, r \in R_s^*$.

We have $t(Other(Q)) = p(Other(Q)) = s \in S$

$t(Q.\langle selector\rangle.\langle attrs\rangle) = p(Q.\langle selector\rangle.\langle attrs\rangle) = r(Q.\langle selector\rangle.\langle attrs\rangle)$

$t(F_i[A_j]) = F_i[P^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p)\, \forall i = 1, ..., k$.

By Lemma 2, we have $t(F_i[A_j]) = F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t)\, \forall i = 1, ..., k$.

Let $t' = t(Q.\langle selector\rangle)$ then

$t'(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle) = r(Q.\langle selector\rangle.\langle attrs\rangle)$

and $t'(F_i[A_j]) = F_i[P^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](p) = F_i[R_s^*; Q\langle gb\_attr\rangle; A_j](r)\, \forall i = 1, ..., k$ so $t' \in R_s$.

Clearly $t = (t', s) \Rightarrow t \in E so E' \subseteq E$ (*)

b) Now we have to show that $E \subseteq E'$

$\forall t \in E \Rightarrow t = (r, s)$ where $s \in S$, $r \in R_s$.

Since $r \in R_s$ then $\exists t' \in R_s^*$, $r(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle)$ and $r(F_i[A_j]) = F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t')\ \forall i = 1, ..., k$.

Let $p = (t', s)$, we have $p \in P^*$ and

$t(Q.\langle selector\rangle.\langle attrs\rangle) = r(Q.\langle selector\rangle.\langle attrs\rangle) = t'(Q.\langle selector\rangle.\langle attrs\rangle)$
$= p(Q.\langle selector\rangle.\langle attrs\rangle)$

$t(F_i[A_j]) = r(F_i[A_j]) = F_i[R_s^*; Q.\langle gb\_attr\rangle; A_j](t')\ \forall i = 1, ..., k$

By Lemma 2, we have

$t(F_i[A_j]) = F_i[P^*; Q\langle gb\_attr\rangle \cup Other(Q); A_j](p)\ \forall i = 1, ..., k$ so $t \in E' \Rightarrow E' \subseteq E$ (**)

From (*) and (**) $\Rightarrow E = E'$ □

By Theorem 1, we have the following remark:

Remark 1. Let Q be a query of the form

SELECT $\langle selector\rangle$ FROM $\langle relation\_list\rangle$ WHERE $\langle predicate\rangle$ GROUP BY $\langle gb\_attr\rangle$
HAVING $\langle hav\_condition\rangle$

then Q is equivalent to the query Q' of the form

SELECT $Q.\langle selector\rangle \cup Other(Q)$ FROM $Q.\langle relation\_list\rangle \cup rels(Other(Q))$
WHERE $Q.\langle predicate\rangle$ GROUP BY $Q.\langle gb\_attr\rangle \cup Other(Q)$ HAVING $Q.\langle hav\_condition\rangle$.

Theorem 2. *Let Q be a query of the form*

SELECT $\langle selector\rangle$ FROM $\langle relation\_list\rangle$ WHERE $\langle predicate\rangle$ GROUP BY $\langle gb\_attr\rangle$
HAVING $\langle hav\_condition\rangle$

then we have

$$Q = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists r(R^*(r) \wedge t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = r(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) \wedge (t(F_i[A_j]) = F_i[R^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](r)\ i = 1, ..., k))\}$$

where $Q.\langle ngb\_query\rangle^*$ denotes the query of the form

SELECT * FROM $Q.\langle relation\_list\rangle$ WHERE $Q.\langle predicate\rangle$

$R = car(meaning(Q.\langle ngb\_query\rangle^*) \cup (rels(Other(Q)) - rels(meaning(Q.\langle ngb\_query\rangle^*))$

$R^* = \{r \mid R(r) \wedge \forall r'(R(r') \wedge (r'(Q.\langle gb\_attr\rangle \cup Other(Q)) = r(Q.\langle gb\_attr\rangle \cup Other(Q)) \rightarrow H(r')))\}$.

Theorem 2 allows to express queries with GROUP BY clause by the result of queries without GROUP BY clause.

The proof of Theorem 2 is based on the Lemma 3. As above, we omit here the proof of Lemma 3.

Lemma 3. *We have $R = \{p \mid P(p) \wedge F(p)\}$.*

Proof of Theorem 2.

By Lemma 3 we have $R^* = P_{FH}$.

By Theorem 1 clearly

$$Q = \{t(Q.\langle selector\rangle \cup Other(Q)) \mid \exists r(R^*(r) \wedge t(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) = r(Q.\langle selector\rangle.\langle attrs\rangle \cup Other(Q)) \wedge (t(F_i[A_j]) = F_i[R^*; Q.\langle gb\_attr\rangle \cup Other(Q); A_j](r)\ i = 1, ..., k))\}.$$ □

By Theorem 2, we have the following remark:

Remark 2. Let Q be a query of the form

SELECT $\langle selector \rangle$ FROM $\langle relation\_list \rangle$ WHERE $\langle predicate \rangle$ GROUP BY $\langle gb\_attr \rangle$ HAVING $\langle hav\_condition \rangle$

then the following queries are equivalent:

i) Q

ii) Q' = SELECT $Q.\langle selector \rangle \cup Other(Q)$ FROM TEMP1 GROUP BY $Q.\langle gb\_attr \rangle \cup Other(Q)$,
where TEMP1 = $meaning(Q^*)$
with Q^* = SELECT * FROM $Q.\langle relation\_list \rangle$ WHERE $Q.\langle predicate \rangle$ GROUP BY $Q.\langle gb\_attr \rangle$ HAVING $Q.\langle hav\_condition \rangle$.

iii) Q'' = SELECT $Q.\langle selector \rangle \cup Other(Q.\langle ngb\_query \rangle)$ FROM TEMP2 GROUP BY $Q.\langle gb\_attr \rangle$ $\cup Other(Q.\langle ngb\_query \rangle)$ HAVING $Q.\langle hav\_condition \rangle$,
where TEMP2 = $meaning(Q.\langle ngb\_query \rangle^*)$.

REFERENCES

[1] S. Ceri, G. Gottlob, Translating SQL into Relational Algebra: Optimization, Semantics, and Equivalence of SQL Queries, *IEEE Trans. Comput.*, Vol. SE-11, No. 4 (1985) 324-345.

[2] C. J. Date., *An introduction to Database System*, 2nd ed., Addision-Wesley Publishing Company 1997. (The translation to Vietnamese by Ho Thuan, Nguyen Quang Vinh, Nguyen Xuan Huy).

[3] R. F. Lans., *Introduction to SQL*, Addision-Wesley Publishing Company, 1988.

[4] J. Paredaens, P. D. Bra, M. Gyssen, D. V. Gucht., *The structure of the Relational Database Model*, Springer Verlag, 1989.

[5] J. Ulman, *Principles of Database Systems*, Computer Science Press, 1980.

Tóm tắt. Trong bài báo này chúng tôi trình bày một số kết quả về sự tương đương giữa những câu hỏi của SQL và biểu thức trong phép tính quan hệ, một số tính chất của câu hỏi trong SQL. Những kết quả này được sử dụng cho việc chuyển dịch câu hỏi của SQL vào đại số quan hệ.

Received: March 11, 1998

(1) Institute of Information Technology.

(2) Pedagogical Institute of Quy Nhon.