关系数据库 外联接 等价变换 逻辑分析 18

73-76

外联接的等价变换与逻辑分析

Equivalence Transformation of Outer-join and Its Logic Analysis

张二忠 楼荣生

（复旦大学计算机科学系 上海 200433）

TP311.13

摘 要 In relational database, outer join is an important operation in which all tuples of the operands are reserved, and present DBMSs only support SQL outer-join. By introducing the method of Extended Three Valued Predicate Calculus of [3], this paper discussed the equivalence transformation of SQL and classical outer-join, and analysed them in logic.

关键词 SQL Language, Outer join, Logic analysis, SQL, Query transformation

在关系数据库中，外联接是在允许空值情况下很重要的操作，Oracle，Sybase 等数据库管理系统能够支持 SQL 外联接操作。由于目前对 SQL 外联接和经典外联接研究还较少，因而限制了它们的使用。

本文从 SQL 外联接与经典外联接的比较入手，对 SQL 外联接与经典外联接进行了一系列的等价变换。然后，在文[3]提供的 E3VPC 方法的基础上，引入了空值处理，对 SQL 外联接与经典外联接从逻辑角度进行了分析。

一、SQL 外联接与经典外联接的关系

Oracle 的 SQL * PLUS 中给出外联接的最简单形式为：

select * from R, S where R.a=S.c(+)；

文[2]将 SQL 外联接限定为下列形式：

select * from $R_1, \cdots, R_n$ where p_1 and … and p_k；

记 $N=\{R_1,\cdots,R_n\}$，表示参加外联接的表集，$P=\{p_1,\cdots,p_k\}$ 表示外联接的联接条件集，且 $p_i(i=1,\cdots,k)$ 是形如 $R_i.a=R_j.b(+)$ 或 $R_i.a(+)=R_j.b$ 的外联接谓词。

下面给出 SQL 外联接的形式定义：

定义 1 设 R，S 是具有公共属性集 Y 的两个关系，R 和 S 中的剩余属性分别为 X 和 Z，用 $R\sim S\times\varnothing_s$ 表示 R 相对于 S 的悬浮行（R～S）与 S 的一个空值元组（$\varnothing_s$）的笛卡尔积组成的关系，则 R 和 S 的 SQL 外联接（记为 R©S）定义为：

$$R©S=R\infty S\cup R\sim S\times\varnothing_s \qquad (1)$$

为了便于说明，本文引入二个关系（基本表）：R(a，b)和 S(c，d)作数值举例，且有值为：

R	a	b	S	c	d
	1	1		2	2
	2	2		3	3
	3			4	
		4			5

其中假定属性 b、c 类型相同。

对定义 1，b、c 为联接属性，R、S 中的剩余属性分别为 a 和 d，则 R©S 结果为：

R©S	a	b	c	d
	1	1		
	2	2	2	2
	3			
		4	4	

文[1]给出经典外联接的定义：

定义 2 设 R，S 是具有公共属性 Y 的两个关系，设 R 和 S 中剩余的属性集分别为 X 和 Z，$R\sim S\times\varnothing_s$ 表示 R 相对于 S 的悬浮行（R～S）与 S 的一个空值元组（$\varnothing_s$）的笛卡尔积组成的关系，用 $\varnothing_R\times S\sim R$ 表示 R 的一个空值元组（$\varnothing_R$）与 S 相对于 R 的悬浮行（S～R）的笛卡尔积组成的关系，则 R 和 S 的外联接（记作 R®S）定义为：

$$R®S=(R\infty S)\cup R\sim S\times\varnothing_s\cup\varnothing_R\times S\sim R \qquad (2)$$

R®S 结果为：

R®S	a	b	c	d
	1	1		
	2	2	2	2
	3			
		4	4	
			3	3
				5

在 R©S 中，只包括 R 中的悬浮元组，而 R®S 中不仅包括 R 的悬浮元组，而且还列出了 S 的悬浮元组。

二、SQL 外联接与经典外联接的等价变换

2.1 SQL 外联接的等价变换

由 R©S 定义知：

$R©S = R\infty S \cup R \sim S \times \varnothing_s$

$= R\infty S \cup (R - \pi_R(R\infty S)) \times \varnothing_s$

它用普通 SQL 语句表示为：

```
select *
from R,S
where r.b=s.c
Union
select r.a,r.b,null,null
from R
where not exists(select *
                 from S
                 where r.b=s.c);                (3)
```

该语句中，使用了 Union 操作和 not exists 子查询。Union 之前的语句为 R 和 S 的自然联接，得到的元组为(2,2,2,2)及(ω,4,4,ω)；Union 之后的语句为 R 的悬浮元组与 S 的空元组的笛卡尔积，使 Union 两端有相同的属性列。其结果为：(1,1,ω,ω)及(3,ω,ω,ω)。

在 Union 操作之后，NULL 列被 S 的列代替。整个语句的结果与(1)式结果相同。

2.2 经典外联接的等价表示

经典外联接不仅包含了自然联接部分，而且还包含了每个关系的悬浮元组。由 R®S 定义知：

$R®S = (R\infty S) \cup R \sim S \times \varnothing_s \cup \varnothing_R \times S \sim R = R\infty S \cup ((R - \pi_R(R\infty S)) \times \varnothing_s) \cup (\varnothing_R \times (S - \pi_S(R\infty S)))$

用普通 SQL 语句表示为：

```
select *
from R,S
where r.b=s.c
Union
select r.a,r.b,null,null
from R
where not exists(select *
                 from S
                 where r.b=s.c)
Union
select null,null,s.c,s.d
from S
where not exists(select *
                 from R
                 where r.b=s.c);                (4)
```

该表达式使用了两个 Union 操作，而且使用了两个 not exists 子查询，查询结果与(2)式结果相同。

经典外联接还可以用 SQL 外联接表示为：

```
select * from R,S where r.b=s.c(+)
Union
select * from R,S where r.b(+)=s.c;
```

三、E3VPC 的引入与扩展

3.1 E3VPC 的引入

E3VPC 是扩展的三值谓词演算表达式，这里只给出主要的结构，详见文[3]。E3VPC 的基本结构为：

$\{t(v1,\cdots,vn):\|P(v1,\cdots,vn)\|^{\alpha}\}$

其中 v1，…，vn 为元组变量，t(v1，…，vn)为目标元组，P(v1，…，vn)为谓词公式。$\|\cdots\|^{\alpha}$ 为把 Unknown 解释成 α 的解释操作符，α 为一布尔值。

令 P(x)为三值谓词公式，Q(x)为二值谓词公式，则有如下定义：

定义3 Q(x)为 P(x)的真解释二值等式，如果对所有 x 成立，下式：

$$Q(x)=\begin{cases}T & 当\ P(X)=U\\ P(X) & 其他\end{cases}$$

记为 $Q(x)=\|P(x)\|^{T}$。

定义4 Q(x)为 P(x)的假解释二值等式，如果对所有 x 成立，下式：

$$Q(x)=\begin{cases}F & 当\ P(X)=U\\ P(X) & 其他\end{cases}$$

记为 $Q(x)=\|P(x)\|^{F}$。

在定义3中，P(x)为 Unknown 时解释为 T；在定义4中，P(x)为 Unknown 时解释为 F。

文[3]中 SQL 查询语句转换为如下形式：

$\{TR\langle FRCLAUSE\rangle:\|TR\langle WHCLAUSE\rangle \wedge \langle HCLAUSE\rangle\|^{\alpha}\}$ (5)

其中，FRCLAUSE，WHCLAUSE 和 HCLAUSE 分别对应于 SQL 中的 FROM 子句，WHERE 子句和 HAVING 子句。下面以引理形式列出文[3]对上述解释算符的推导规则，证明可参见原文。

引理1 设 P、Q 是三值谓词，α∈{F,T}。

1) $\|P(x)\vee Q(x)\|^{\alpha}\Leftrightarrow\|P(x)\|^{\alpha}\vee\|Q(x)\|^{\alpha}$

2) $\|P(x)\wedge Q(x)\|^{\alpha}\Leftrightarrow\|P(x)\|^{\alpha}\wedge\|Q(x)\|^{\alpha}$

3) $\|\neg P(x)\|^{\alpha}\Leftrightarrow\neg\|P(x)\|^{\neg\alpha}$

4) $\|\|P(x)\|^{\alpha}\|^{\beta}\Leftrightarrow\|P(x)\|^{\alpha}$

5) $\|\exists\ x\ in\ R:P(x)\|^{\alpha}\Leftrightarrow\exists\ x\ in\ R:\|P(x)\|^{\alpha}$

6) $\|\forall\ x\ in\ R:P(x)\|^{\alpha}\Leftrightarrow\forall\ x\ in\ R:\|P(x)\|^{\alpha}$

7) $\|\exists\ \{x\ in\ R:\|P(x)\|^{\alpha}\}Q(x)\|^{\beta}\Leftrightarrow\exists\ x\ in\ R:\|P(x)\|^{\alpha}\wedge\|Q(x)\|^{\beta}$

8) $\|\forall\ \{x\ in\ R:\|P(x)\|^{\alpha}\}Q(x)\|^{\beta}\Leftrightarrow\forall\ x\ in\ R:\|\neg P(x)\|^{\neg\alpha}\vee\|Q(x)\|^{\beta}$

3.2 E3VPC 中空值的引入

文[3]中并未提供关于 NULL 的查询语义。这里将 NULL 查询语义加入到查询条件子句中：

⟨NULL PRED⟩::=⟨COL⟩IS [NOT]NULL

因而也加入相应的解释语句：

$\langle COL\rangle = [<>]_\omega$

其中 ω 表示空值。下文将研究外联接条件 r.b=s.c(+)的表达方法。

四、外联接的逻辑表示

4.1 SQL 外联接的逻辑表示

(5)式中不包含 select 子句，为体现 select r.a，r.b，null，null，(3)式用 E3VPC 表示为：

$\{r \text{ in } R, s \text{ in } S: \|r.b=s.c\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\}$ (6)

该表达式由两个集合组成，"∪"前面部分表示 Union 前面的自然联接，"∪"后面的部分表示 R 的悬浮元组与 S 的空元素的笛卡尔积。令 S'=S∪{∅ₛ}，且根据引理1中的推导规则，(6)式可合并为：

$\{r \text{ in } R, s \text{ in } S': (\|r.b=s.c\|^F \vee \forall s \text{ in } S: \|r.b<>s.c\|^T \wedge s=\varnothing_s\}$

其中$\|r.b=s.c\|^F$ 表示条件 r.b=s.c 有空值时为假；而$\|r.b<>s.c\|^T$ 表示条件 r.b<>s.c 有空值时为真。

在 R©S 产生的结果中，元组(2,2,2,2)和(ω,4,4,ω)是由条件$\|r.b=s.c\|^F$ 取出的；元组(1,1,ω,ω)和(3,ω,ω,ω)是由条件：$\|r.b<>s.c\|^T$ 取出的。R 中的属性 b 为 ω，ω 与 S 中任一值比较皆为 Unknown，Unknown 在$\|r.b=s.c\|^F$ 中为假，而在$\|r.b<>s.c\|^T$ 中为真。

4.2 经典外联接的逻辑表示

(4)式用 E3VPC 表示为：

$\{r \text{ in } R, s \text{ in } S: \|r.b=s.c\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\} \cup \{\varnothing_R, s \text{ in } S: \|\neg\exists \{r \text{ in } R: \|r.b=s.c\|^F\}\|^F\}$

令 S'=S∪{∅ₛ}，R'=R∪{∅ᵣ}，则上式合并为：

$\{r \text{ in } R', s \text{ in } S': \|r.b=s.c\|^F \vee (\forall s \text{ in } S: \|r.b<>s.c\|^T) \wedge s=\varnothing_s \vee (\forall r \text{ in } R: \|r.b<>s.c\|^T) \wedge r=\varnothing_R\}$

因此可得出 SQL 外联接与经典外联接的 E3VPC 表达式，即引理2。

引理2 关系 R 和 S 的 SQL 外联接和经典外联接各表示为 R©S 和 R®S，则：

$R©S=\{r \text{ in } R, s \text{ in } S': (\|r.b=s.c\|^F \vee \forall s \text{ in } S: \|r.b<>s.c\|^T \wedge S=\varnothing_s\}$

$R®S=\{r \text{ in } R', s \text{ in } S': \|r.b=s.c\|^F \vee (\forall s \text{ in } S: \|r.b<>s.c\|^T) \wedge s=\varnothing_s \vee (\forall r \text{ in } R: \|r.b<>s.c\|^T) \wedge r=\varnothing_R\}$

证：把 $R©S=\{r \text{ in } R, s \text{ in } S: \|r.b=s.c\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\}$

$R®S=\{r \text{ in } R, s \text{ in } S: \|r.b=s.c\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\} \cup \{\varnothing_R, s \text{ in } S: \|\neg\exists \{r \text{ in } R: \|r.b=s.c\|^F\}\|^F\}$

的集合并运算转换成逻辑运算即可。

五、外联接等价变换的逻辑分析

5.1 悬浮元组的逻辑分析

2.1节中的(3)式中悬浮元组部分的 E3VPC 表达式为：

$\{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\}$ (7)

该悬浮元组用带有外联接与 NULL 的表达式为：

```
select *
from R,S
where r.b=s.c(+) and s.c is null[或 s.d is null]
```

该式省去了 not exists 操作，由引理2的结论与3.2节对 E3VPC 的扩展，用 E3VPC 将其表示为：

$(\{r \text{ in } R, s \text{ in } S: \|r.b=s.c\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\}) \cap \{r \text{ in } R, s \text{ in } S: \|s.c=\omega\|^F\}$ (8)

定理1 在两个基本表中，带一个外联接表示的悬浮元组与带 not exists 表示的悬浮元组等价。

证：该定理即证(7)与(8)等价。

(8)式 $=\{r \text{ in } R, s \text{ in } S: (\|r.b=s.c\|^F)\} \cap \{r \text{ in } R, s \text{ in } S: \|s.c=\omega\|^F\} \cup \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\} \cap \{r \text{ in } R, s \text{ in } S: \|s.c=\omega\|^F\} = \{r \text{ in } R, s \text{ in } S: \|r.b=s.c \wedge s.c=\omega\|^F\} \cup \{r \text{ in } R, s \text{ in } S': \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F \wedge \|s.c=\omega\|^F \wedge s=\varnothing_s\}$
$=\{r \text{ in } R, s \text{ in } S': \|r.b=s.c \wedge s.c=\omega\|^F \wedge s=\varnothing_s \vee \forall s \text{ in } S: \|r.b<>s.c\|^T \wedge \|s.c=\omega\|^F \wedge s=\varnothing_s\} = \varnothing \cup \{r \text{ in } R, \varnothing_s: \forall s \text{ in } S: \|r.b<>s.c\|^T\} = \{r \text{ in } R, \varnothing_s: \|\neg\exists \{s \text{ in } S: \|r.b=s.c\|^F\}\|^F\} =$ (7)式

条件 r.b=s.c 表示在 R,S 的笛卡尔积中取满足 r.b=s.c 且 s.c<>ω 的元组，而条件 s.c=ω 表示在 R,S 中取满足 s.c=ω 的元组，结果为∅。

$\|r.b<>s.c\|^T = \neg\|r.b=s.c\|^F$，而$\|r.b=s.c\|^F$ 为 T 时 s.c 不空，而$\|s.c=\omega\|^F$ 为 T 时 s.c 空，所以条件$\|s.c=\omega\|^F$ 取出的元组必包含于条件$\neg\|r.b=s.c\|^F$ 取出的元组。

5.2 多个外联接

作为应用实例，分析一个较复杂的例子。

例1

```
select r.a,r.b
from R,S
where r.b =s.c(+)and s.d(+)
        =2 and s.c is null;
```

将外联接表示为 E3VPC 表达式。因外联接实际由两部分组成：一部分为自然联接的结果；一部分为悬浮元组与一关系空元组的笛卡尔积。

Where 条件中的 $s.d(+)=2$，若右边为一个列名是不允许的，因为这里的 S 表上有两个外联接符号[2]，但本条件中等号右边是常数，该条件等价于：在已有的外联接中再与虚表"2"作外联接，即 $s.d<>2$ 或 $s.d$ 为空的那些行也令 $s=\varnothing_s$。因此，在结果中的行有三类：一是 $r.b=s.c$ 且 $s.d=2$ 的行；二是 $r.b=s.c$ 但 $s.d<>2$ 的行，同时有 $s=\varnothing_s$；三是不存在 s 使 $r.b=s.c$ 的行，同时也有 $s=\varnothing_s$。再结合第三个条件 s.c is null，查询的 E3VPC 表达式可写为：

$$\{r \text{ in } R, s \text{ in } S' : (\|r.b=s.c \wedge s.d=2\|^F) \vee (\forall \{s \text{ in } S : \|r.b=s.c\|^F\}\|s.d<>2\|^T) \wedge s=\varnothing_s \vee (\forall s \text{ in } S : \|r.b<>s.c\|^T) \wedge s=\varnothing_s) \wedge \|s.c=\omega\|^F\} = \{r \text{ in } R, s \text{ in } S' : (\forall s \text{ in } S : \|r.b<>s.c\|^T \vee \|s.d<>2\|^T) \wedge \|s.c=\omega\|^F \wedge s=\varnothing_s \vee (\forall s \text{ in } S : \|r.b<>s.c\|^T) \wedge \|s.c=\omega\|^F \wedge s=\varnothing_s\} = \{r \text{ in } R, s \text{ in } S' : (\forall s \text{ in } S : \|r.b<>s.c \vee s.d<>2\|^T) \wedge \|s.c=\omega\|^F \wedge s=\varnothing_s)\} = \{r \text{ in } R, s \text{ in } S' : \neg\exists\ s \text{ in } S : \|r.b=s.c \wedge s.d=2\|^F \wedge s=\varnothing_s\} = \{r \text{ in } R, \varnothing_s : \neg\exists\ s \text{ in } S : \|r.b=s.c \wedge s.d=2\|^F\} = \{r \text{ in } R : \neg\exists\ s \text{ in } S : \|r.b=s.c \wedge s.d=2\|^F\}$$

写出 SQL 的等价语句为：

```
select r.a,r.b
from R
where not exists(select *
                 from S
                 where r.b=s.c and s.c=2)
```

结束语 本文通过扩展三值谓词演算方法，分析了外联接语义，对查询条件形如 $a=b(+)$ 的 SQL 语句进行了形式化表示，同时对外联接与带 not exists 子查询之间的转换关系进行了初步的探讨。实际中，如果能用外联接代替 not exists 子查询将会加快查询速度。因此，进一步的研究是有意义的。

下一步的工作将进一步研究含有外联接的查询语句的等价变换问题。其中 E3VPC 是一个可用的工具。

参考文献

[1]施伯乐、何继潮、崔靖. 关系数据库理论和应用. 河南科学技术出版社, 1989

[2]楼荣生. SQL 外联接及性质. 广西师范大学学报.(第六届全国数据库教学会议论文集)

[3]M. Negri. Formal Semantics of SQL Queries. ACM Trans. Database Syst.,(9)1991

[4]丁宝康. SQL 语言中谓词和空值的使用. 计算机研究与发展,(3)1994

[5]Oracle 关系数据库用户手册

(上接第102页)

3. 软件开发是一件创造性的活动，不可能为一些严格的规则和子过程所约束，PSEEs 必须能支持过程执行者的活动而不以某些指定的行为予以强迫。如 PSEEs 环境向设计者提供支持，让他能指定应用的体系结构，但不强迫他使用某指定的工具或方法；

4. 软件的生产通常是由开发者在不同的场点进行的，因此有必要建立 PSEEs 的分布式联邦。所谓联邦，就是提供弱的完整性约束，改善并行性和独立性，但是分散的团队能协同工作；

5. 各种工具集成起来向用户、设计者提供更有效的软件开发环境，PSEEs 必须能控制工具的执行并检查它们的操作是否和过程模型所定义的一致。

主要参考文献

[1]G. E. Kaiser et al., Intelligent assistance for software development and maintenance. IEEE Software, May 1988

[2]M. M. Lehman. Software Engineering, the Software Process and their Support. Soft. Eng. J., No. 5 1991

[3]I. R. Mcchesney. Toward a Classification Scheme for Software Process Modelling Approaches. Infor. and Software Tech., No. 7 1995

[4]A. Fuggetta. Functionality and Architecture of PSEEs. Infor. and Software Tech., 1996

[5]W. L. Melo et al., A Software Engineering Environment Driven by E-C-A Rules and its Trigger Mechanism. Infor. and Software Tech., No. 10 1995