Bases de Datos
Práctico 6: Optimización en el Plan Lógico de Consultas

G. Saiz - N. Wolovick

4 de octubre de 2000

1. Para las siguientes relaciones

$$\begin{aligned} r(AB) &= \{(0,1),(2,3),(0,1),(2,4),(3,4)\} \\ s(AB) &= \{(0,1),(2,4),(2,5),(3,4),(0,2),(3,4)\} \end{aligned}$$

   computar

   (a) $r \cap_B s$
   (b) $r -_B s$
   (c) $\Pi_{a+b,b^2}(r)$
   (d) $r \bowtie_{R.b<S.b} s$
   (e) $\gamma_{a,SUM(b)}(r)$
   (f) $\delta(r)$
   (g) $\tau_{b,a}(r)$

Operadores Join-Like

Existen varios operadores del estilo del join que son provistos por bases de datos comerciales.

- El *semijoin* de las relaciones r y s $r \ltimes s$, es el multiconjunto de tuplas t tal que hay al menos una tupla en s que concuerda con t en todos los atributos comunes de R y S.
- El *antisemijoin* $r \overline{\ltimes} s$, es la bolsa de tuplas t en r que no concuerdan con ninguna tupla de s en los atributos $R \cap S$.
- El *outerjoin* $r \overset{\circ}{\bowtie} s$ se forma con $r \bowtie s$ más el agregado de las *dangling tuples* de r o s, donde una tupla está colgando si esta no se junta con ninguna de la otra relación. Las tuplas agregadas se rellenan con el símbolo de indefinido o nulo $\perp$, en aquellos atributos que la tupla colgante no posee pero que si aparecen en la relación resultante.
- El *leftouterjoin* $r \overset{\circ}{\bowtie}_L s$ es como el *outerjoin*, pero solo las tuplas colgantes de r son rellenadas con $\perp$ y agregadas al resultado.
- El *rightouterjoin* $r \overset{\circ}{\bowtie}_R s$ también es como el *outerjoin*, pero esta vez el argumento que genera las *dangling tuples* es el derecho.

2. Dar expresiones para los cinco operadores definidos en el párrafo anterior usando solamente los operadores standard del álgebra relacional. Para las variantes "*outerjoin*" puede usar una **relación nula** $n(A_1, \ldots, A_i)$ que consiste de una única tupla con $\perp$ en cada componente.

3. Un operador unario f es *idempotente* si, $\forall r f^2(r) = f(r)$. Para cada uno de los operadores δ, Π_L, σ_C, γ_L, τ, explique por que son idempotentes o muestre un contraejemplo.

4. Cuando es posible empujar una selección en los dos argumentos de un operador binario, es necesario decidir si hacerlo o no. ¿Cómo afectaría la existencia de índices en uno de los argumentos?. Considere, por ejemplo, una expresión $\sigma_C(r \cap s)$, donde se tiene un índice sobre s.

5. Dar ejemplos donde se muestre que la eliminación de duplicados no puede ser empujada por debajo de la unión o diferencia de multiconjuntos.

6. Los operadores similares al *join* del ejercicio 2 obedecen algunas reglas comunes y otras no. Para cada caso pruebe o de un contraejemplo para las siguientes leyes.

   - $\sigma_C(r \ltimes s) = \sigma_C(r) \ltimes s$
   - $\sigma_C(r \dot{\bowtie} s) = \sigma_C(r) \dot{\bowtie} s$
   - $\Pi_L(r \overline{\ltimes} s) = \Pi_L(r) \overline{\ltimes} s$
   - $r \ltimes s = s \ltimes r$

7. Reemplace los join naturales en las expresiones siguientes por theta-joins y proyecciones. Diga cuando los theta-join resultantes forman un grupo asociativo y conmutativo.

   (a) $(r(AB) \bowtie s(BC)) \bowtie_{S.c > T.c} t(CD)$
   (b) $(r(AB) \bowtie s(BC)) \bowtie (t(CD) \bowtie u(DE))$
   (c) $(r(AB) \bowtie s(BC)) \bowtie (t(CD) \bowtie u(AD))$

8. Dar reglas para convertir cada una de las siguientes formas de condición en álgebra relacional. Asuma que las condiciones se aplican a una relación $r(R)$ y no están correlacionadas a ella. Tenga especial cuidado para no introducir o eliminar duplicados respecto a la definicion formal de SQL.

   - `EXISTS` <*query*>
   - $a =$ `ANY` <*query*>, donde a es un atributo de R
   - $a =$ `ALL` <*query*>, donde a es un atributo de R

9. Transforme la siguiente query en un plan lógico y trate de mejorarlo aplicando leyes algebraicas.

```
SELECT starName
FROM Actor1996
WHERE studioName="Paramount";
```

donde hay dos vistas definidas

```
CREATE VIEW Actor1996 AS
       SELECT starName, studioName
       FROM MoviesOf1996 NATURAL JOIN starsIn;

CREATE VIEW MovieOf1996 AS
       SELECT *
       FROM Movie
       WHERE year=1996;
```

y los esquemas son

```
Movie(title,year,length,inColor,studioName,producerC#)
StarsIn(title,year,starName)
```