Cálculo Relacional de Tuplas

El cálculo relacional de tuplas es un lenguaje declarativo que permite especificar, por comprensión, las propiedades que deben poseer las tuplas de un conjunto. En tal sentido se dice que es lenguaje orientado a tuplas ya que para poder determinar el conjunto resultado se requiere el procesamiento tupla a tupla de las relaciones que permiten obtener las tuplas del conjunto resultado.

A diferencia del álgebra relacional, la cual visualiza las operaciones de una forma procedimental (un requerimiento debemos expresarlo a través de las operaciones que deben realizarse para resolverlo), el cálculo relacional sólo permite indicar cuál es el requerimiento y no cómo debe ser llevado a cabo. Por tal fin no es necesario ofrecer una serie de operaciones y un lenguaje de formación de operaciones sino ofrecer un lenguaje de especificación declarativa de cuál es la operación a resolver.

Por estas características es sencillo apoyarse en una extensión a la lógica tradicional adaptada al modelo relacional que permite llevar a cabo tal especificación.

Así, toda expresión del cálculo relacional tiene la forma

$$\{t_1.A_1,t_2.A_2,t_3.A_3,....,t_n.A_n \mid CONDICION(t_1,t_2,...,t_n,t_{n+1},...,t_{n+m}\}$$

donde:

- Cada t_i $1 \le i \le m+n$ recibe el nombre de **variable de tuplas**. Como su nombre lo indican permiten representar las tuplas sobre las cuales se desea trabajar. Sin embargo toda tupla pertenece a una relación. Llamaremos **relación rango** a la relación en la cual tiene dominio una variable de tupla.
- Cada A_i asociado a un t_i a través del operador punto (.) permite indicar el valor del atributo de nombre A_i en la tupla t_i. Claro esta que el atributo A_i debe ser atributo de la relación rango de t_i.
- $CONDICION(t_1,t_2,...,t_n,t_{n+1},...,t_{n+m})$ es una expresión en una lógica de primer orden extendida para el modelo relacional que se explicará luego.

Significado: La expresión

$$\{t_1.A_1,t_2.A_2,t_3.A_3,....,t_n.A_n \mid CONDICION(t_1,t_2,...,t_n,t_{n+1},...,t_{n+m})\}$$

expresa el requerimiento: "Dar los valores de $t_1.A_1$,..., $t_n.A_n$ tales que se cumple la condición $CONDICION(t_1,t_2,...,t_n,t_{n+1},...,t_{n+m})$.

Nótese que los t_i no necesariamente tienen que ser diferentes.

Veamos ahora el lenguaje a utilizar para definir condiciones. Este lenguaje, como se mencionó anteriormente, no es más que una extensión a la lógica de primer orden para poder manejar relaciones. El lenguaje a utilizar se basa en el concepto de fórmula como un grupo de átomos: Un átomo puede ser:

- Un predicado de la forma R(t) donde R es el nombre de una relación de la base de datos y t es una variable de tuplas. Con esta especificación se indica que R es la relación rango de la variable de tuplas t.
- Una expresión de la forma t_i.A <operador> t_j.B. En este caso se quiere indicar la comparación por <operador> del valor del atributo A en la tupla t_i con el valor del atributo B de la tupla t_j. Obviamente debe cumplirse que el atributo A sea un atributo de la relación rango de t_i y el atributo B sea atributo de la relación rango de t_j. Es fundamental (y puede constituirse como un error) que a toda variable de tupla se le indique la relación rango en la condición. Los operadores permitidos son $=,\neq,<,\leq, >$ y $\geq$.
- Una expresión de la forma t_i.A <operador> <constante> o t_i.A <operador> <constante>. En este caso se quiere indicar la comparación por <operador> del valor del atributo A en la tupla t_i con una constante dada. Obviamente debe cumplirse que <valor> se encuentre en el dominio del atributo A. Los operadores permitidos son $=,\neq,<,\leq,>$ y $\geq$.

A partir del concepto de átomo podemos construir fórmulas como:

- Todo átomo es una fórmula.
- Si F_1 y F_2 son fórmulas, entonces F_1 and F_2, F_1 or F_2 y not F_1.
- Si F_1 es una fórmula y t es una variable de tuplas,$\exists t(F_1)$ es una fórmula (cuantificación existencial de t)
- Si F_1 es una fórmula y t es u,$\forall t(F_1)$ es una fórmula (cuantificación universal de t).

Finalmente debe tenerse en cuenta el hecho de que toda variable de tuplas debe ocurrir ligada en la condición. Por defecto, toda variable de tupla indicada como resultado ocurre ligada en la condición si se indica la relación rango. Las variables de tuplas no declaradas en el resultado y usadas en la condición ocurrirán ligadas si y sólo si se indica su relación rango y se cuantifican universal o existencialmente en la condición.

Ejemplos:

- El requerimiento "Dar la cédula de identidad de las personas que sintonizan al menos la emisora X o la emisora Y, podría expresarse como:

  {t_1.CI | SINTONIZA(t_1) and (t_1.CodEm = 'X' or t_1.CodEm = 'Y')}

- El requerimiento "Dar la cédula de identidad de las personas que sintonizan al menos la emisora 'X' y la emisora 'Y'", podría expresarse como:

$\{t_1.CI \mid SINTONIZA(t_1)$ and $t_1.CodEm =$ 'X' and
$\exists t_2(SINTONIZA(t2)$ and $t_1.CI = t2.CI$ and $t2.CodEM =$ 'Y')}

- El requerimiento "Dar las personas (cédula de identidad y nombre) que sólo sintonizan la emisora 'X' y la emisora 'Y'", podría expresarse como:

$\{t_1.CI, t_1.Nombre \mid PERSONA(t_1)$ and
$\exists t_2(SINTONIZA(t_3)$ and $t_2.CI = t_1.CI$ and $t_2.CodEm =$ 'X') and
$\exists t_3(SINTONIZA(t_3)$ and $t_3.CI = t_1.CI$ and $t_3.CodEm =$ 'Y') and
not $\exists t_4(SINTONIZA(t_4)$ and $t_4.CI = t_1.CI$ and
$(t_4.CodEm \neq X$ and $t_2.CodEm \neq Y))\}$

- El requerimiento "Dar el código de las emisoras donde no se colocan las canciones 'Z' ni 'W' " podría expresarse como:

$\{t.CodEm \mid EMISORA(t)$ and not $\exists t_1(COLOCA(t_1)$ and $t_1.CodEm = t.CodEm$ and
$(t_1.CodCan =$ 'Z' or $t_1.CodCan =$ 'W'))}

- "Dar el nombre de las canciones que le gustan a José Pérez".

$\{t_1.Nombre \mid CANCION(t_1)$ and $\exists t_2 \exists t_3(GUSTA(t_2)$ and $PERSONA(t_3)$ and
$t_2.CodCan = t_1.CodCan$ and $t_2.CI = t_3.CI$ and $t3.Nombre =$ 'José Pérez')}

- "Dar el nombre de los radioescuchas que sintonizan sólo emisoras que colocan la canción 'W' ".

$\{t_1.Nombre \mid PERSONA(t_1)$ and
not $\exists t_2$ $(COLOCA(t_2)$ and $t_2.CodCan =$ 'W' and
$\forall t_3(SINTONIZA(t_3)$ and $t_3.CI=t_1.CI$ and
$t_3.CodEm = t_2.CodEm))\}$

EXPRESIONES SEGURAS

La gramática del lenguaje de especificación del cálculo relacional no prohíbe la ocurrencia de situaciones como not EMISORA(x). El significado de esta fórmula quiere indicar cualquier tupla que no sea emisora. Sin embargo cómo trabajar con un dominio infinito de tuplas? Esta noción se escapa a lo que es como tal computación. Para tal fin se definen lo que se conocen como expresiones (reglas) seguras. En cálculo relacional, se dice que una expresión es segura si:

- No ocurren átomos de la forma not R(t), con t una variable de tupla ligada.
- Todas las variables de tuplas ocurren ligadas en la fórmula.
- Si un atributo A ocurre sobre un dominio infinito y es utilizado en una variable de tupla cuantificada universalmente, no pueden ocurrir operadores que incluyan negación (no igual, no menor, no mayor, entre otros).

En general, una regla semántica del cálculo relacional determina que una expresión en cálculo **no** es correcta si alguna de las fórmulas utilizadas en ella no es segura.

En general, no es sencillo transformar una fórmula no segura en una fórmula segura. Algunas reglas de interés para poder hacer esta transformación son:

- $\forall x(P(x)) \equiv \text{not } (\exists x(\text{not } P(x))$
- $\exists x(P(x)) \equiv \text{not } (\forall x(\text{not } P(x)))$
- $\forall x(P(x) \text{ and } Q(x)) \equiv \text{not } (\exists x(\text{not } P(x) \text{ or not } Q(x)))$
- $\forall x(P(x) \text{ or } Q(x)) \equiv \text{not } (\exists x(\text{not } P(x) \text{ and not } Q(x)))$
- $\exists x(P(x) \text{ or } Q(x)) \equiv \text{not } (\forall x(\text{not } P(x) \text{ and not } Q(x)))$
- $\exists x(P(x) \text{ and } Q(x)) \equiv \text{not } (\forall x(\text{not } P(x) \text{ or not } Q(x)))$

Veamos algunos ejemplos:

- Supongamos el requerimiento. Dar el nombre las emisoras que son escuchadas por todos los radioescuchas.

Supongamos que tuviésemos la implicación para expresar este requerimiento. Entonces tendríamos:

$$\{t_1.\text{Nombre} \mid \text{EMISORA}(t_1) \text{ and } \forall t_2(\text{PERSONA}(t_2) \rightarrow \exists t_3(\text{SINTONIZA}(t_3) \text{ and } t_3.CI = t_2.CI \text{ and } t_3.\text{CodEm} = t_1.\text{CodEm}))\}$$

Sabemos que la implicación $P(x) \rightarrow Q(x)$ puede ser expresada como (not $P(x)$ or $Q(x)$), entonces podemos expresar el requerimiento así:

$\{t_1.Nombre \mid EMISORA(t_1) \text{ and } \forall t_2(\text{not } PERSONA(t_2) \text{ or } \exists t_3(SINTONIZA(t_3) \text{ and } t_3.CI = t2.CI \text{ and } t_3.CodEm = t_1.CodEm))\}$

Así, obtenemos una regla insegura (debido a not $PERSONA(t_2)$). Apliquemos la regla

$$\forall x(P(x) \text{ or } Q(x)) \equiv \text{not } (\exists x(\text{not } P(x) \text{ and not } Q(x)))$$

para hacerla segura:

Sean:

- $F_1(t_1,t_2) = (\text{not } F_2(t_2) \text{ or } F_3(t_1,t_2))$
- $F_2(t_2) = \text{not } PERSONA(t_2)$
- $F_3(t_1,t_2) = \exists t_3(SINTONIZA(t_3) \text{ and } t_3.CI = t2.CI \text{ and } t_3.CodEm = t_1.CodEm)$

Así $\forall t_2(F_1(t_1,t_2)) \equiv \forall t_2(\text{ not } F_2(t_2) \text{ or } F_3(t_1,t_2)) \equiv \text{not } \exists t2(\text{not } (\text{not } F_2(t_2) \text{ or } F_3(t_1,t_2)) \equiv$
$\equiv \text{not } \exists t_2(F_2(t_2) \text{ and not } F_3(t_1,t_2))$

De esta forma el requerimiento, expresado por una fórmula segura sería:

$\{t_1.Nombre \mid EMISORA(t_1) \text{ and not } \exists t_2(PERSONA(t_2) \text{ and not } \exists t_3(SINTONIZA(t_3) \text{ and } t_3.CI = t_2.CI \text{ and } t_3.CodEm = t_1.CodEm))\}$