

Resueltos de la auxiliar de Álgebra Relacional

Esquemas

Los esquemas son los siguientes:

PC(PC_Id, nombre, precio, cantidad)

Componente(Com_Id, nombre, precio)

PCComp(PC_Id, Com_Id)

(8) Los diez componentes más baratos

Veamos cómo abordar este problema. Determinemos algunas propiedades: un componente está entre los 10 más baratos $\Rightarrow$ hay, a lo más, 10 componentes más baratos o al mismo precio (contándose a sí mismo). Entonces, los siguientes problemas son equivalentes:

“Obtener los N componentes más baratos”

$\Leftrightarrow$ “Calcular cuántos componentes son más baratos que uno dado”

¿Y cómo vemos cuántos componentes son más baratos que un componente dado? Hay que comparar la relación *Componente* consigo misma. En otras palabras, hay que hacer un producto cruz de *Componente* con *Componente*.

Luego, la solución es:

$$A, B \leftarrow \text{Componente}$$

$$C \leftarrow \sigma_{A.\text{precio} \geq B.\text{precio}}(A \times B)$$

$$D \leftarrow \langle A.Com_Id \rangle \bowtie_{\langle A.Com_Id, count(B.Com_Id) as N \rangle} (C)$$

$$Resp \leftarrow \sigma_{N \leq 10}(D)$$

(9) Los componentes usados en todos los computadores

Esto se puede leer básicamente como “los componentes que se relacionan con todos los computadores.” Y como sabemos, esto es una división:

$$Resp \leftarrow PCComp \div \pi_{\{PC_Id\}}(PC)$$

También se puede resolver sin división, pero la solución obedecería la definición relacional de división a menos que se use agrupación y agregación.

(10) Los computadores que tienen un componente que cuesta el doble de algún otro componente del equipo

Para responder a esta pregunta basta encontrar un par de componentes que formen parte del mismo computador y que uno tenga el doble del precio del otro. Luego, a través de saber a qué computador pertenecen, se contesta la consulta.

Entonces obtengamos los pares de componentes de un mismo computador

$$A \leftarrow (\rho_{\{PC_Id as X, Com_Id as C1\}}(PCComp)) \bowtie_{X=Y} (\rho_{\{PC_Id as Y, Com_Id as C2\}}(PCComp))$$

luego obtenemos la información de los precios de cada componente

$$B \leftarrow \pi_{\{P1, C1\}}(\rho_{\{Com_Id as C1, precio as P1\}}(Componente))$$

$$C \leftarrow \pi_{\{C2, P2\}}(\rho_{\{Com_Id as C2, precio as P2\}}(Componente))$$

$$D \leftarrow (\pi_{\{X, C1, C2\}}(A) \bowtie B) \bowtie C$$

y obtenemos un esquema del tipo $(X, C1, P1, C2, P2)$ con Ck un *Com_Id* y Pk un *precio*.

Finalmente queda exigir $P1 = 2 \times P2$ y proyectar por X , que es un *PC_Id*. Pero haremos un join con *PC* para hacer más claro el resultado:

$$Resp \leftarrow \pi_{esq(PC)}(PC \bowtie_{X=PC_Id} (\sigma_{P1=2P2}(D)))$$

donde proyectamos por *esq(PC)* para deshacernos de los términos innecesarios.

(11) De todos los computadores, la mitad más barata

Este problema es equivalente al problema (8), pero en vez de exigir $N \leq 10$ exigimos $N \leq \frac{M}{2}$, con M el número de [modelos de] computadores, y tratamos con computadores en vez de componentes. Obviamente, M debe obtenerse con agregación.

Otra manera de resolverlo, y la que se presentará acá, es exigir que, para cierto computador dado, haya más computadores más caros que computadores más baratos. Explotando esta propiedad, construimos una crusa de la misma relación tres veces:

$$A, B, C \leftarrow PC$$

$$D \leftarrow \sigma_{A.PC_Id=B.PC_Id \wedge B.PC_Id=C.PC_Id}(A \times B \times C)$$

que deja un esquema $D(A.PC_Id, A.nombre, A.precio, A.cantidad, B.PC_Id, B.nombre, \dots)$.

Ahora, exijamos que para cada tupla de A , las tuplas de B tienen un precio mayor o igual y las de C un precio menor o igual:

$$E \leftarrow \sigma_{A.precio \leq B.precio \wedge A.precio \geq C.precio}(D)$$

y obtenemos $N1$ el número de tuplas tan o más caras y $N2$ el número de tuplas tan o más baratas:

$$F \leftarrow \langle A.PC_Id \rangle \bowtie_{\langle A.PC_Id, count(B.PC_Id) \text{ as } N1, count(C.PC_Id) \text{ as } N2 \rangle} (E)$$

lo que nos arroja un esquema $F(A.PC_Id, N1, N2)$, suficiente para concluir:

$$Resp \leftarrow PC \bowtie \sigma_{N1 \geq N2}(F)$$

(12) Los pares de computadores que están hechos exactamente de los mismos componentes

Esto no es división, pero se parece mucho en el sentido de que se buscan los pares de computadores que tienen exactamente las mismas relaciones. Eso sí, tener las mismas relaciones implica que no se pueden escapar relaciones a considerar en el *match*.

Este problema puede ser contraintuitivo si se considera el estilo de resolución de problemas usado hasta ahora, pero es bastante sencillo de resolver. En efecto, sólo hay que contar: si el número de componentes en común para un par de computadores es igual al número de componentes de cada computador, necesariamente ese par de computadores está hecho de los mismos componentes. (Si cree que no, demuestre esta proposición usando contradicción para convencerse.)

Primero construyamos el número de componentes por computador:

$$A, B \leftarrow \langle PC_Id \rangle \bowtie_{\langle count(*) \text{ as } N, PC_Id \rangle} PCComp$$

y construyo los pares de computadores para un mismo componente, sólo para agrupar y contar:

$$C, D \leftarrow PCComp$$

$$E \leftarrow \langle C.PC_Id, D.PC_Id \rangle \bowtie_{\langle count(*) \text{ as } N, C.PC_Id \text{ as } AId, D.PC_Id \text{ as } BId \rangle} (\sigma_{C.Com_Id = D.Com_Id}(C \times D))$$

y verifico que los componentes comunes a los pares sean igual a los componentes de cada computador, concluyendo:

$$Resp \leftarrow \pi_{\{E.AId, E.BId\}} \sigma \left\{ \begin{array}{l} A.PC_Id = E.AId \quad \wedge \quad B.PC_Id = E.BId \quad \wedge \\ A.N = E.N \quad \wedge \quad B.N = E.N \end{array} \right\} A \times B \times E$$

Propuesto: Construya la división usando el operador $\bowtie$ de agrupación y agregación. Note que la

división no usa “exactamente igual” sino “al menos.”

(13) Los pares de computadores que tienen al menos tres componentes iguales

Primero generamos tríos de componentes, todos distintos, que pertenecen a un mismo computador. Luego, comparamos dos tríos y verificamos que los componentes sean iguales pero los computadores diferentes:

$$A, B, C \leftarrow PCComp$$

$$D \leftarrow \sigma \left(\begin{array}{l} A.Com_Id \neq B.Com_Id \quad \wedge \quad A.Com_Id \neq C.Com_Id \quad \wedge \quad B.Com_Id \neq C.Com_Id \\ \wedge \quad A.PC_Id = B.PC_Id \quad \wedge \quad B.PC_Id = C.PC_Id \end{array} \right)^{(A \times B \times C)}$$

y renombramos convenientemente:

$$E \leftarrow \rho \left\{ \begin{array}{l} A.PC_Id \text{ as } PC_Id, \\ A.Com_Id \text{ as } C1, \\ B.Com_Id \text{ as } C2, \\ C.Com_Id \text{ as } C3 \end{array} \right\} \pi \left\{ \begin{array}{l} A.PC_Id, \\ A.Com_Id, \\ B.Com_Id, \\ C.Com_Id \end{array} \right\}^D$$

Ahora comparemos los tríos para concluir:

$$F \leftarrow E$$

$$Resp \leftarrow \pi_{\{E.PC_Id, F.PC_Id\}} \sigma \left(\begin{array}{l} E.PC_Id \neq F.PC_Id \\ \wedge E.C1 = F.C1 \\ \wedge E.C2 = F.C2 \\ \wedge E.C3 = F.C3 \end{array} \right)^{(E \times F)}$$

(Este problema es más sencillo de resolver usando el estilo del problema anterior. La solución elegida fue usada para ilustrar otra manera de resolver el problema. En general, existen muchas maneras de resolver un problema. Con un poquito de ingenio se pueden obtener muchas más. Y para reforzar el ingenio, nada mejor que ver varias maneras de resolver un problema.

En los problemas 9, 11 y 13 se indica explícitamente cómo es una de las soluciones alternativas. Trate de resolverlos de esa manera para reforzar su habilidad.)