43.75% han aumentado los impuestos y para el 18.75 % realmente no han tenido una incidencia significativa (Ver Figura 12).

CONCLUSIONES Y RECOMENDACIONES

1. Aunque el gobierno permite calcular la depreciación por el sistema de Línea Recta, por el de Reducción de Saldos o por otro que exista en cualquier país del mundo de reconocido valor técnico, autorizado previamente por la Dirección General de Impuestos Nacionales, se evidencia una marcada tendencia a la utilización del método de Línea Recta (LR), debido a su tradicionalidad en el régimen contable, fácil cálculo y por el desconocimiento de otros métodos.
2. Se evidencia un marcado desconocimiento de las empresas en el tema de los beneficios económicos, financieros y tributarios a que puede conllevar el implementar un determinado sistema de depreciación.
3. En la gran empresa del Valle del Cauca, se presenta una notoria tendencia a medir los efectos de la depreciación desde un enfoque eminentemente fiscal.
4. Dado que cada empresa posee sus propias necesidades y estrategias de mercado, se requiere estudiar su situación particular para poder elegir un método de depreciación que contribuya al alcance de los objetivos en el mediano y corto plazo.

BIBLIOGRAFIA

1. Varela, Rodrigo. *Evaluación económica de inversiones,* Editorial Norma, 5ª Edición, Bogotá, 1994.
2. Varela, Rodrigo. *Los Sistemas de depreciación bajo los ajustes integrales por Inflación,* ICESI, Cali, 1995.
3. Varela, Rodrigo. *Cambio de sistema de depreciación bajo los ajustes integrales por inflación,* ICESI, Cali, 1995.
4. Acuña, Melquisedec, Giraldo Carlos. *Efectos fiscales de los ajustes integrales por inflación en la evaluación financiera de proyectos,* Facultad de Ciencias de la Administración, Cali, 1994.
5. Legis. *Régimen del Impuesto a la Renta y Complementarios,* Legis Editorial S.A., Bogotá, 1993.
6. Legis. *Ajustes integrales por inflación,* Legis Editorial S.A., Bogotá, 1991.
7. Impuestos 1996. *Nueva Reforma Tributaria, Ley 223 de 1995.* Tomos I y II, Centro Interamericano Jurídico-Financiero, Santafé de Bogotá, 1995.

UN DISEÑO OPTIMO EN TERCERA FORMA NORMAL

LUIS EDUARDO MUNERA

Matemático de la Universidad del Valle, Máster y Doctor en Informática de la Universidad Politécnica de Madrid. Ex profesor de la Facultad de Informática de la Universidad Politécnica de Madrid. Profesor ICESI.

1. INTRODUCCION

Existen métodos clásicos de diseño de bases de datos relacionales –descomposición y síntesis– El objetivo de esas aproximaciones clásicas es alcanzar el más alto nivel de normalización posible.

El método de diseño por descomposición es "top-down" que comienza con una relación existente, investiga su forma normal y la descompone vía proyecciones hasta que el esquema relacional adquiera el grado de normalización deseado.

El método de diseño por síntesis es una aproximación "bottom-up" la cual comienza con un conjunto de dependencias no redundantes y combina los atributos involucrados en esas dependencias, en una forma normal (usualmente la tercera forma normal).

Se han propuesto varios algoritmos de diseño por síntesis, pero los más conocidos son los de Ullman[5] y Bernstein.[2]

El algoritmo de Ullman se destaca por su sencillez y facilidad de uso, además que garantiza la conservación de dependencias y la propiedad Lossless-join. Sin embargo, tiene dos inconvenientes serios. El primero es que posee un criterio parcial de detección de esquemas en tercera forma normal que sólo le permite detectarlos en el caso de que en el esquema $R < T,L >$ exista una dependencia en L, de la forma $X \rightarrow A$, tal que $X \cup \{ A \} = T$.

Esto tiene el inconveniente que el algoritmo no va a detectar la mayoría de casos de esquemas en tercera forma normal, produciendo una descomposición innecesaria de esos esquemas.

El segundo es que el algoritmo crea un esquema para cada dependencia funcional de L, generando una gran cantidad de esquemas.

El algoritmo de Bernstein se destaca porque genera un número mucho menor de esquemas.

Sin embargo, también presenta varios inconvenientes. En primer lugar, puede fallar al detectar esquemas en tercera forma normal. Por ejemplo, si tenemos un esquema $R < T, L>$ con $T = \{A, B, C, D, E, G. H\}$ y $L = \{A \rightarrow B, B \rightarrow C, CG \rightarrow A, CG \rightarrow D, CG \rightarrow E, CG \rightarrow H\}$,

el resultado de aplicar el algoritmo de Bernstein es el diseño, ρ = {R1, R2, R3} con T1 = {A, B}, T2 = {B, C} y T3 = {A, C, D, E, G, H} con llaves k1 = {A}, K2 = {B} y K3 = {CG}; descomposición que es innecesaria porque el esquema original se encuentra en tercera forma normal.

En segundo lugar, aunque una de las fortalezas del algoritmo es que genera un número reducido de esquemas, que en muchos casos es el mínimo, hay casos en los que no garantiza un conjunto mínimo de esquemas en tercera forma normal. Por ejemplo, si tenemos un esquema R < T, L > con T = {A, B, C, D, X, Y} y L = {XY→ AD, CD → YX, AX →B, BY→C, C→A}, el algoritmo de Bernstein genera el diseño ρ = {R1, R2, R3, R4} con T1 = {C, D, X, Y}, T2 = {A, X, B}, T3 = {B, Y, C} y T4 = {C, A} con llaves K1 = {CD, XY}, K2 = {AX}, K3 = {BY}, K4 = {C}. Este conjunto se puede reducir aún más; tanto CD como XY son llaves del esquema inicial, por lo tanto C, D, X, Y son atributos principales y por ende puedo agrupar los esquemas R1 y R3 sin dañar la tercera forma normal, pues C es principal, produciéndose un diseño más óptimo ρ' = {R1', R2', R3'} con T 1' = {C, D, X, Y, B}, T_2 = {A, X, B} y T3 = {C, A} con K1' = {CD, XY), K2' = {AX} y K3' = {C}.

También hay casos en los que el algoritmo de Bernstein genera diseños que no satisfacen la propiedad "Loss-less-join".

Teniendo en cuenta todo lo anterior, podemos señalar que el objetivo de este trabajo es proponer un algoritmo de diseño de síntesis, que genere un diseño óptimo, entendiendo por óptimo lo siguiente:

1. El número de esquemas que se generan es mínimo.
2. Los esquemas que se generan están en tercera forma normal de CODD.
3. Se satisface la propiedad "Loss-less-join".
4. No hay pérdida de dependencias.

2. DEFINICIONES BASICAS

El diseño de síntesis de bases de datos relacionales se trabaja partiendo de un esquema universal U < T, L >, en donde T es un conjunto de atributos y L es un conjunto de dependencias funcionales, X→Y, siendo X,Y⊆T, llamados descriptores. Cada dependencia funcional de L es una restricción que señala que a cada valor asociado a los atributos de X, les corresponde uno y sólo un valor asociado a los atributos de Y.

Las dependencias funcionales satisfacen un conjunto de axiomas definidos por Armstrong.[1]

1. REFLEXIVIDAD: X→X, X ⊆ T
2. AUMENTATIVIDAD: SI X→Y, ENTONCES X'→ Y', X' ⊇ X
3. PROYECTIVIDAD: SI X→Y, ENTONCES X→ Y', Y'⊆Y
4. ADITIVIDAD: SI X→Y, M→ N, ENTONCES X∪M→Y∪N
5. TRANSITIVIDAD: SI X→Y, Y→Z, ENTONCES X→Z.

Definición: Denominamos cierre de L^+, a L unido con el conjunto de dependencias funcionales que se pueden obtener a partir de L por aplicación de los axiomas de Armstrong.

Definición: Sea un descriptor X ⊆ T y conjunto L de dependencias funcionales. El cierre de X respecto de L, X^+, es el descriptor formado por los atributos A ∈ T, tal que (X→ A) ∈ L^+.

Definición: Un descriptor K ⊆T es llave de un esquema R < T, L > si K' = T y no existe un descriptor K' ⊂ K tal que $(K')^+$ = T.

En Fernández[3] se encuentra un algoritmo que permite calcular el cierre de un descriptor, también se encuentra un algoritmo que permite calcular todas las llaves asociadas a un esquema R < T, L >.

Definición: Dados dos conjuntos de dependencias funcionales L y M, decimos que son equivalentes, si y sólo si, $L^+ = M^+$.

Definición: Dado un conjunto M de dependencias funcionales, decimos que es no redundante, cuando:

1. Todas sus dependencias son de la forma X → A (segundos miembros simples).
2. No hay dependencias redundantes. Una dependencia (X → A) ∈ M es redundante en M cuando su supresión no altera el cierre, o sea:

   (M - {X A}) = M^+.
3. No hay atributos extraños. Un atributo B ∈ X es extraño en la dependencia X → A de M cuando, llamando Z al descriptor X - {B}, el cierre de M no se altera al sustituir X → A por Z → A, o sea: (M - {X → A} U {Z → A})$^+$ = M^+.

Dado un conjunto L de dependencias funcionales, siempre existe un conjunto M equivalente y no redundante. Este conjunto M no redundante se puede calcular mediante algunos algoritmos de recubrimientos que se pueden encontrar en Ullman[5] y Maier.[4]

Definición: La descomposición de un esquema de relación R< T, L>, es una sustitución por un conjunto de proyecciones ρ = {R1, R2, ..., Rk} tal que T1∪ T2 ∪ ... ∪ Tk = T y los Ti no son necesariamente disjuntos.

Definición: Sea ρ = {R1, R2, ..., Rk} una descomposición del esquema R< T, L>. Sea Ri < Ti, Li > la especificación completa del esquema i-ésimo. Decimos que ρ es una descomposición con la propiedad de unión sin pérdida de información (propiedad Lossless-join) respecto de L si, para toda ocurrencia r de R (para toda asignación de valores a los atributos de T), se verifica la igualdad:

r = π T1 (r) |x| π T2 (r) |x| ... |x| π Tk (r) que significa la posibilidad de reconstruir "r" a partir de la join de sus proyecciones.

Definición: Sea el esquema R < T, L > con L = { X → Y / (X ∪ Y) ⊆ T} y sea ρ = {R1, R2, ..., Rk} una descomposición con Ri < Ti, Li >. Decimos que Li es la proyección de L sobre Ti, Si: Li = {(Xi → Yi) ∈ L^+ / (Xi ∪ Yi) ⊆Ti} y decimos que ρ preserva las dependencias si

$$(\cup_{i=1}^{k} Li)^+ = L^+$$

3. ALGORITMO DE DISEÑO

Entrada: Un esquema R < T, L > en donde M es un conjunto no redundante, y un conjunto K de todas las claves del esquema.

Salida: Una descomposición ρ de R < T, L > con un número mínimo de esquemas, cada uno de los cuales está en 3FN y tal que conserva las dependencias y satisface la propiedad Loss-less-join.

METODOS

1. Si para cada dependencia X → A ∈ M se cumple que X contiene una clave o A pertenece a alguna clave, entonces R < T, M > está en 3FN y Fin del algoritmo.

   En caso contrario:
2. Partir M en grupos tales que todas las dependencias de cada grupo tengan los mismos implicantes.
3. Para cada grupo construir un esquema Ri < Ti, Li > en donde Ti es el conjunto de todos los atributos del grupo y Li es la proyección de M sobre Ti.

   Incluir los Ri en ρ.
4. Para cada dos esquemas Ri < Ti, Li > y Rj < T j, Lj > en ρ con conjuntos

de claves Ki y Kj, respectivamente, tales que existe una clave Ci $\in$ Ki y una clave Cj $\in$Kj, tal que Ci $\leftrightarrow$ Cj. Entonces podemos agrupar los dos esquemas en un nuevo esquema Rij = < Tij, Lij > tal que Tij = Ti $\cup$ Tj y Lij es la proyección de M sobre Tij.

SI Rij está en 3 FN (Aplicamos el paso 1 sobre Lij teniendo en cuenta un Kij). Entonces borramos Ri y Rj en ρ e incluimos Rij en ρ. En caso contrario, calculamos un recubrimiento Mij de Ljk, de la siguiente manera: Si existe una dependencia X $\rightarrow$ A $\in$ E Lij que es transitiva, la eliminamos de Lij junto con toda otra dependencia de la forma Y $\rightarrow$ A $\in$ Lij. También eliminamos el atributo A en Tij.

Obtenemos un nuevo esquema R'ij < T'ij, L'ij > en donde Tij' = Tij - {A}. Luego borramos Ri y Rj en ρ e incluimos R'ij.

Repetir hasta que no haya esquemas en ρ que cumplan con esas características.

5. Para cada dos esquemas Ri < Ti, Li > y Rj < Tj, Lj > en ρ con conjuntos de claves de Ki y Kj respectivamente, tales que existe una clave Ci $\rightarrow$ Cj, y todos los atributos de Ti - Cj son principales en R < T, M >, entonces podemos agrupar los dos esquemas en un nuevo esquema Rij < Tij, Lij > tal que Tij = Ti $\cup$ Tj y Lij es la proyección de M sobre Tij. Luego borramos Ri y Rj en ρ e incluimos Rij.

   Repetir hasta que no haya esquemas en ρ que cumplan con esas características.

6. Si ninguna clave c $\in$ K está incluida en algún Ti asociado a algún Ri $\in$ ρ. Entonces creamos un nuevo esquema Rc = < Tc, Lc > con alguna clave c $\in$ K. En donde, Tc = c, y Lc es un conjunto de dependencias triviales de la forma, c $\rightarrow$ c', $\forall$c' $\subseteq$ c. Por lo que se puede omitir. Incluimos T_C en ρ.

4. EJEMPLO

Sea T = {A, B_1, B_2, C_2, C_1, D, E, I_1, I_2, I_3, J} y

L = {A$\rightarrow$$B_1$ B_2 C_1C_2DE$I_1I_2I_3$J, $B_1B_2C_1\rightarrow$ $AC_2DEI_1I_2I_3J$, $B_1B_2C_2 \rightarrow$ $AC_1DEI_1I_2I_3J$, E$\rightarrow$ $I_1I_2I_3$, $C_1D\rightarrow$ J, $C_2D \rightarrow$ J, $I_1I_2 \rightarrow I_3$, $I_2I_3 \rightarrow I_1$, $I_1I_3 \rightarrow I_2$}

M = {A $\rightarrow$ B_1, A $\rightarrow$ B_2, A$\rightarrow$$C_2$, $B_1B_2C_1 \rightarrow$ A, $B_1B_2C_2 \rightarrow C_1$, $B_1B_2C_2 \rightarrow$ D $B_1B_2C_2$ $\rightarrow$ E, E $\rightarrow$ I_1, E $\rightarrow$ I_2, $C_1D \rightarrow$ J,

$C_2D \rightarrow$ J, $I_1I_2 \rightarrow I_3$,$I_2I_3 \rightarrow I_1$, $I_1I_3 \rightarrow I_2$}

K ={A, $B_1B_2C_1$, $B_1B_2C_2$}

1. El esquema no está en tercera forma normal.
2. M_1 = {A$\rightarrow$ B_1, A $\rightarrow$ B_2, A $\rightarrow$ C_2}, M_2 = {$B_1B_2C_1 \rightarrow$ A}

   M_3 = {$B_1B_2C_2 \rightarrow C_1$, $B_1B_2C_2 \rightarrow$ D, $B_1B_2C_2 \rightarrow$ E}

   M_4 = {E $\rightarrow$ I_1, E $\rightarrow$ I_2},

   M_5 = {$C_1D \rightarrow$ J}, M_6 = {$C_2D \rightarrow$ J},

   M_7 = {$I_1I_2 \rightarrow I_3$}, M_8 = {$I_2I_3 \rightarrow I_1$},

   M_9 = {$I_1I_3 \rightarrow I_2$}.
3. R_1: T_1 = {A, B_1, B_2, C_2}, L_1 = {A $\rightarrow$ B_1, A $\rightarrow$ B_2, A $\rightarrow$ C_2, $B_1B_2C_2 \rightarrow$A} K_1 = {A, $B_1B_2C_2$}.

   R_2: T_2 = {A, B_1, B_2 C_1}, L_2 = {$B_1B_2C_1 \rightarrow$ A, A $\rightarrow$ B_1, A $\rightarrow$ B_2, A $\rightarrow$ C_1},

   K_2 = {A, $B_1B_2C_1$}.

   R_3: T_3 = {B_1, B_2, C_2, C_1, D, E}, L_3 = {$B_1B_2C_2 \rightarrow C_1$, $B_1B_2C_2 \rightarrow$ D, $B_1B_2C_2$ E, $B_1B_2C_1 \rightarrow C_2$, $B_1B_2C_1 \rightarrow$ D, $B_1B_2C_1$ $\rightarrow$ E},

   K_3 = {$B_1B_2C_1$, $B_1B_2C_2$}.

   R_4: T_4 = {E, I_1, I_2}, L_4 = {E $\rightarrow$ I_1, E $\rightarrow$ I_2}, K_4 = {E}.

   R_5: T_5 = {C_1, D, J}, L_5 = {$C_1D \rightarrow$ J}, K_5 = {C_1D}.

   R_6: T_6 = {C_2, D, J}, L_6 = {$C_2D \rightarrow$ J}, K_6 = {C_2D}.

   R_7: T_7 = {I_1, I_2, I_3}, L_7 = {$I_1I_2 \rightarrow I_3$, $I_2I_3 \rightarrow$ I_1, $I_1I_3 \rightarrow I_2$},

   K_7 = {I_1I_2, I_1I_3, I_2I_3}.

   $R_8 = R_7$, $R_9 = R_8 = R_7$.

   ρ = {R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9}
4. Agrupamos R_1 y R_2, produciendo R_{12}:

   T_{12} = {A, B_1, B_2, C_2, C_1} y L_{12} = {A $\rightarrow$ B_1, A $\rightarrow$ B_2, A $\rightarrow$ C_2, A $\rightarrow$ C_1, $B_1B_2C_1$ A, $B_1B_2C_1\rightarrow C_2$, $B_1B_2C_2 \rightarrow$ A, $B_1B_2C_2$ $\rightarrow$ C_1}

   K_{12} = {A, $B_1B_2C_2$, $B_1B_2C_1$}

Este esquema está en tercera forma normal.

Agrupamos R_{12} y R_3, produciendo R_{123}:

T_{123} = {A, B_1, B_2, C_2, C_1, D, E}

L_{123} = {A $\rightarrow$ $B_1B_2C_2C_1$DE, $B_1B_2C_2 \rightarrow$ AC_1DE, $B_1B_2C_1 \rightarrow$ AC_2DE}

K_{123} = {A, $B_1B_2C_2$, $B_1B_2C_1$}.

Este esquema está en tercera forma normal.

Agrupamos R_7, R_8, R_{19} produciendo R_{789}, T_{789} = {I_1, I_2, I_3}, L_{789} = {$I_1I_2 \rightarrow I_3$, $I_2I_3 \rightarrow$ $I_1I_3 \rightarrow$ I_2}, K_{789} = {I_1I_2, I_1I_3, I_2I_3}.

Este esquema esta en tercera forma normal.

ρ = {R_{123}, R_4, R_5, R_6, R_{789}}.

5. Cualquier clave de R_{123} implica cualquier clave de los otros esquemas de Tj, pero Tj - Cj no son principales en ningún caso. Por lo tanto no agrupamos R_{123} con ningún subesquema.

   E $\rightarrow$ $I_1I_2I_3$ pero T_{789} - K_{789} = ϕ. Por lo tanto ρ no cambia.
6. Las claves están incluidas en R_{123}. Por lo tanto el ρ final es:

   ρ = {R_{123}, R_4, R_5, R_6, R_{789}}.

REFERENCIAS

1. ARMSTRONG, W.W. *Dependency structures of data base relationships.* proceedings 1974 IFIP. Congress; North - Holland, Amsterdam (pp. 580-583).
2. BERNSTEIN, P.A. *Synthesizing third normal form relations from functional dependencies.* ACM transactions on Database Systems. Vol. 1 No. 4, 1976.
3. FERNÁNDEZ, C.B. *El modelo relacional de datos: de los fundamentos a los modelos deductivos.* Ediciones Díaz de Santos, Madrid, 1987.
4. MAIER, D. *The theory of relational databases.* London, Pitman, 1983.
5. ULLMAN, J.D. *Principles of database systems.* Computer Science Press. Rockville, Maryland, 1982.