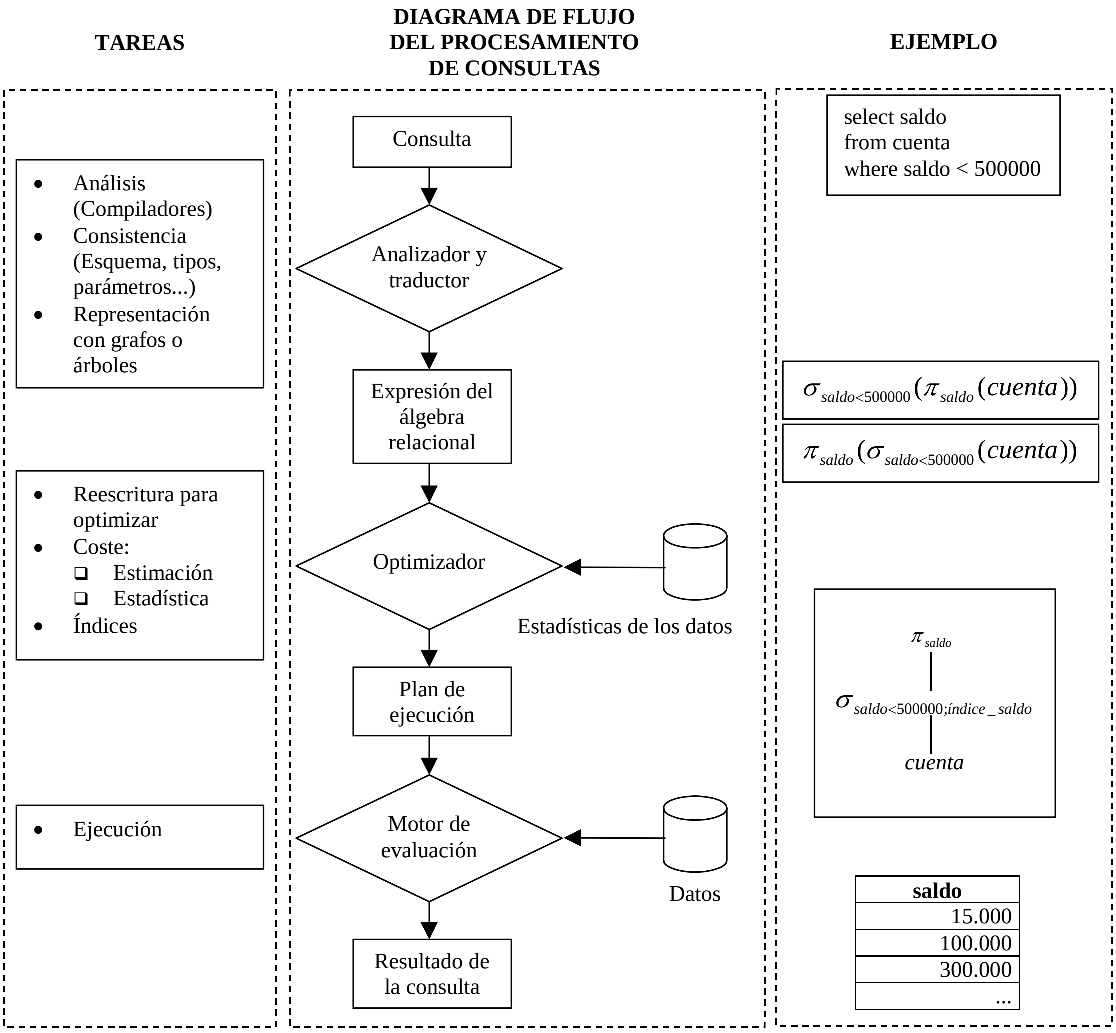
Contenido:

8. Procesamiento de consultas

8.1. Visión general



8.2. Reescritura. Reglas de equivalencia del álgebra relacional

Equivalencia semántica: igual esquema, igual número de tuplas (quizás en diferente orden).
Sean:
θ condición simple
L lista de atributos
R, S y T relaciones
$attrib(X)$ es el conjunto de atributos que aparecen en X.

1) **Operaciones σ, π, $\times$, $\bowtie_\theta$**

1.1) Cascada (secuencia) de σ: $\sigma_{\theta_1 \wedge \ldots \wedge \theta_n}(R) = \sigma_{\theta_1}(\cdots(\sigma_{\theta n}(R)\cdots)$

1.2) Conmutatividad de σ: $\sigma_{\theta_1}(\sigma_{\theta 2}(R)) = \sigma_{\theta 2}(\sigma_{\theta 1}(R))$

1.3) Cascada de π:

i) $\pi_{L_1}(\cdots(\pi_{Ln}(R))\cdots) = \pi_{L_1}(R)$, si $L_1 \subseteq L_2 \subseteq \cdots \subseteq L_n$

ii) $\pi_{L_1}(\cdots(\pi_{Ln}(R))\cdots) = \pi_{L_1 \cap L_2 \cap \cdots \cap L_n}(R)$ en el caso general

1.4) Conmutatividad de σ con π: $\pi_L(\sigma_\theta(R)) = \sigma_\theta(\pi_L(R))$

1.5) Conmutatividad de $\times$: $R \times S = S \times R$

1.6) Conmutatividad de $\bowtie$: $R \bowtie S = S \bowtie R$

1.7) Distributividad de σ con $\bowtie$:

i) $\sigma_\theta(R \bowtie S) = \sigma_\theta(R) \bowtie S$, si $attrib(\theta) \subseteq attrib(R)$

ii) $\sigma_{\theta_1 \wedge \theta_2}(R \bowtie S) = (\sigma_{\theta_1}(R)) \bowtie (\sigma_{\theta_2}(S))$, si

$attrib(\theta_1) \subseteq attrib(R) \wedge attrib(\theta_2) \subseteq attrib(S)$

1.8) Distributividad de σ con $\times$:

i) $\sigma_\theta(R \times S) = \sigma_\theta(R) \times S$, si $attrib(\theta) \subseteq attrib(R)$

ii) $\sigma_{\theta_1 \wedge \theta_2}(R \times S) = (\sigma_{\theta_1}(R)) \times (\sigma_{\theta_2}(S))$, si

$attrib(\theta_1) \subseteq attrib(R) \wedge attrib(\theta_2) \subseteq attrib(S)$

1.9) Distributividad de π con $\times$: $\pi_L(R \times S) = (\pi_{L_1}(R)) \times (\pi_{L_2}(S))$, si

$L_1 \subseteq attrib(R) \wedge L_2 \subseteq attrib(S)$

1.10) Distributividad de π con $\bowtie$:

i) $\pi_L(R \bowtie_\theta S) = (\pi_{L_1}(R)) \bowtie_\theta (\pi_{L_2}(S))$, si $L_1 \subseteq attrib(R) \wedge L_2 \subseteq attrib(S)$

ii) $\pi_L(R \bowtie_\theta S) = \pi_L((\pi_{L_1 \cup C_1}(R)) \bowtie_\theta (\pi_{L_2 \cup C_2}(S)))$, si

$L_1 \subseteq attrib(R) \wedge L_2 \subseteq attrib(S) \wedge$

$C_1 = attrib(\theta) \cap attrib(R) \wedge C_2 = attrib(\theta) \cap attrib(S)$

1.11) Asociatividad de $\times$: $(R \times S) \times T = R \times (S \times T)$

1.12) Asociatividad de $\bowtie$: $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

1.13) Asociatividad de $\bowtie_\theta$: $(R \bowtie_{\theta_{RS}} S) \bowtie_{\theta_{ST} \wedge \theta_{RT}} T = R \bowtie_{\theta_{RS} \wedge \theta_{RT}} (S \bowtie_{\theta_{ST}} T)$, si

$attrib(\theta_{RS}) \subseteq attrib(R) \cup attrib(S) \wedge$

$attrib(\theta_{ST}) \subseteq attrib(S) \cup attrib(T) \wedge$

$attrib(\theta_{RT}) \subseteq attrib(R) \cup attrib(T)$

1.14) Equivalencia con σ, $\times$ y $\bowtie$:

i) $\sigma_\theta(R \times S) = R \bowtie_\theta S$

ii) $\sigma_{\theta 1}(R \bowtie_{\theta 2} S) = R \bowtie_{\theta 1 \wedge \theta 2} S$

2) Operaciones con conjuntos

2.1) Conmutatividad de $\cup$ y $\cap$. – no lo es. $R \cup S = S \cup R$, $R \cap S = S \cap R$

2.2) Asociatividad de $\cup$ y $\cap$.
$(R \cup S) \cup T = R \cup (S \cup T)$, $(R \cap S) \cap T = R \cap (S \cap T)$

2.3) Distributividad de σ con $\cup$: $\sigma_\theta(R \cup S) = (\sigma_\theta(R)) \cup (\sigma_\theta(S))$

2.4) Distributividad de σ con $\cap$ y $-$: Si $\Delta \in \{\cap, -\}$ $\sigma_\theta(R \Delta S) = (\sigma_\theta(R)) \Delta S$

2.5) Distributividad de π con $\cup$: $\pi_L(R \cup S) = (\pi_L(R)) \cup (\pi_L(S))$

8.3. Heurísticas para la optimización

En general se pueden aplicar reglas heurísticas simples tales como:
- Realizar las operaciones σ tan pronto como sea posible.
- Realizar las operaciones π tan pronto como sea posible, pero no antes de las operaciones σ.

Esquema de un algoritmo de optimización algebraico heurístico que incorpora las reglas de equivalencia:

1. Usando la regla 1.1, descomponer las operaciones σ con condiciones conjuntivas en una secuencia de operaciones σ. Permite mayor libertad al mover los nodos σ en el árbol.
2. Usando las reglas 1.2, 1.4, 1.7, 1.8, 2.3 y 2.4 (distributividad de σ con otras operaciones), mover las operaciones σ hacia los nodos hoja tanto como sea posible.
3. Usando las reglas 1.5, 1.6 (conmutatividad), 1.11, 1.12, 1.13 y 2.2 (asociatividad), reorganizar los nodos hoja según:
   - Mover las operaciones σ *más restrictivas* hacia los nodos hoja tanto como sea posible.
   - Asegurarse de que el orden de los nodos hoja no provocan operaciones $\times$.
4. Usando la regla 1.14, combinar la operación $\times$ con una operación σ subsiguiente dando como resultado una operación $\rhd\lhd$.
5. Usando las reglas 1.3, 1.4, 1.10 y 2.5 (secuencias de σ y conmutatividad/distributividad de π con otras operaciones), separar y mover listas de los atributos hacia los nodos hoja tanto como sea posible, creando las operaciones π necesarias.
6. Identificar los subárboles que representan grupos de operaciones que se puedan ejecutar por un único algoritmo.

Ejemplo:
SELECT LNAME
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE PNAME='Aquarius' AND PNUMBER=PNO AND ESSN=SSN AND BDATE>'1957-12-31'

$$\pi_{LNAME}(\sigma_{PNAME='Aquarius' \wedge PNUMBER=PNO \wedge ESSN=SSN \wedge BDATE>'1957-12-31'}(EMPLOYEE \times WORKS\_ON \times PROJECT) =$$

$$\pi_{LNAME}(\sigma_{c_1 \wedge c_2 \wedge c_3 \wedge c_4}(E \times W \times P)),$$

$$c_1 = PNAME = 'Aquarius',$$

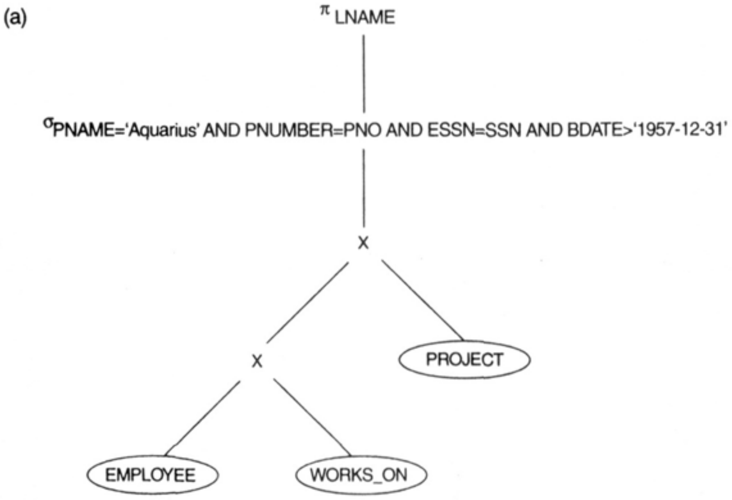
$$c_2 = PNUMBER = PNO,$$

$$c_3 = ESSN = SSN,$$

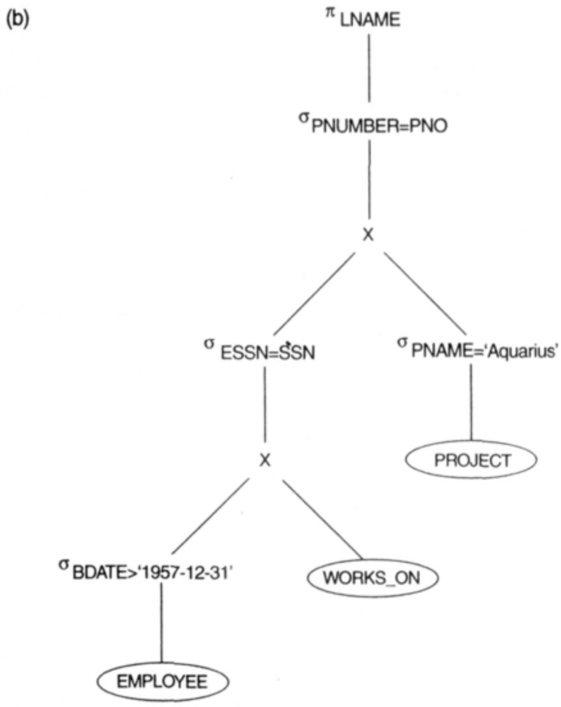
$$c_4 = BDATE > '1957-12-31',$$

$$E = EMPLOYEE, W = WORKS\_ON, P = PROJECT$$

$$\pi_{LNAME}(\sigma_{c_1 \wedge c_2 \wedge c_3 \wedge c_4}(E \times W \times P))$$

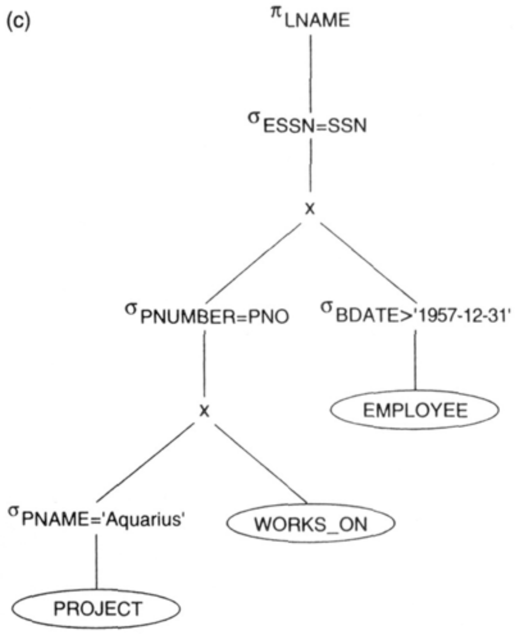


Con los pasos 1 y 2 del algoritmo se pasa a (b):



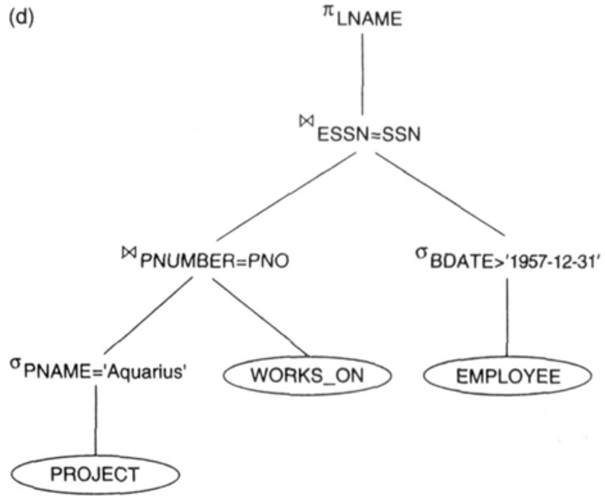
$$\pi_{LNAME}(\sigma_{c_2}(\sigma_{c_3}(\sigma_{c_4}(E) \times W) \times \sigma_{c_1}(P)))$$

Con el paso 3 se pasa a (c):



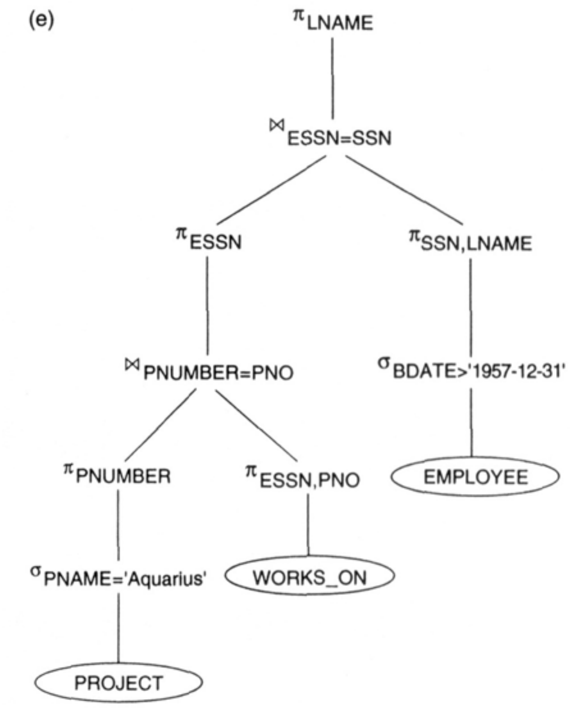
$\pi_{LNAME}(\sigma_{c_3}(\sigma_{c_2}(\sigma_{c_1}(P)\times W)\times\sigma_{c_4}(E)))$

Con el paso 4 se pasa a (d)



$\pi_{LNAME}((\sigma_{c_1}(P)\bowtie_{c_2} W)\bowtie_{c_3}(\sigma_{c_4}(E)))$

Aplicando el paso 5 se pasa a (e):



8.4. Coste

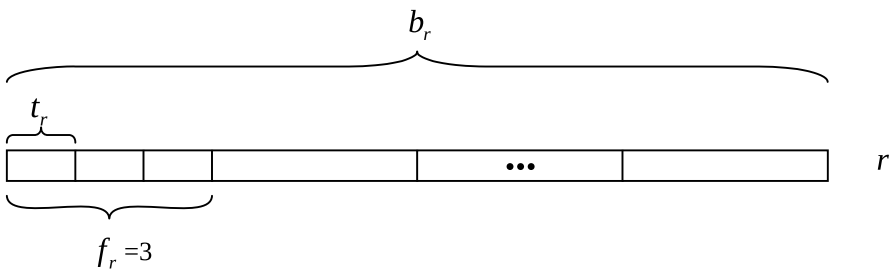
Estimación del coste. Atributos, medidas y métricas.

8.4.1. Atributos

Atributos que determinan la estimación del coste [Def. atributo: Pressman]:

Atributos para relaciones:

- n_r número de tuplas de la relación r.
- b_r número de bloques que contienen tuplas de la relación r.
- t_r tamaño en bytes de una tupla de la relación r.
- f_r factor de bloqueo de la relación r = número de tuplas por bloque
- $V(A,r) = |\pi_A(r)|$ número de valores distintos del atributo A en la relación r.
  - $V(A,r) = n_r$ si A es un atributo clave.
- $CS(A,r)$ es la cardinalidad media de la selección del atributo A en la relación r.
  - $CS(A,r) = 1$ si A es un atributo clave.
  - $CS(A,r) = \dfrac{n_r}{V(A,r)}$ para una distribución uniforme. Se pueden elegir otras distribuciones.



Se cumple $b_r = \left\lceil \frac{n_r}{f_r} \right\rceil$ si las tuplas se almacenan contiguas físicamente.

Idealmente las medidas de los atributos se deben realizar cada vez que se modifique una tabla. Como esto es costoso, se hace en los intervalos de menos carga del sistema.

Atributos para los índices:

- g_i grado de salida de los nodos internos del índice i (para índices con estructura de árbol).
- AA_i el el número de niveles del índice i para la relación r.
  - $AA_i = 1$ para tablas hash
  - $AA_i = \left\lfloor \log_{g_i}(V(A,r)) \right\rfloor$

8.4.2. Medidas

- Tiempo de acceso a disco
- Tiempo de UCP
- Tiempo de comunicación (SGBDs distribuidos o paralelos)

Cuantitativamente:	Tiempo de acceso a disco >	Tiempo de UCP
Realización de la medida:	Fácil	Difícil

Suposición: todas las transferencias de bloques a disco tienen el mismo coste (ignora la latencia rotacional de disco y el tiempo de búsqueda).
Por lo tanto, la medida se toma como el número de transferencias de bloques.

8.5. Ejecución

Las diferencias entre estrategias diferentes son de varios órdenes de magnitud.
Para todos: $E_{Ai} \equiv$ coste del algoritmo A_i

8.5.1. Ordenación externa

- ORDER BY
- JOIN, INTERSECTION
- Eliminación de duplicados en la selección y proyección
- [Wirth94]

8.5.2. Selección

Suposición: Archivo -> relación, registro -> tupla.
1 Condición de igualdad
A1- Búsqueda lineal. $E_{A1} = b_r$ Se recorren todos los bloques

A2- Búsqueda binaria. $E_{A2} = \lceil \log_2 b_r \rceil + \left\lceil \frac{CS(A,r)}{f_r} \right\rceil - 1$

$\lceil \log_2 b_r \rceil$ coste de localizar la primera tupla (número de bloques accedidos).

$\left\lceil \frac{CS(A,r)}{f_r} \right\rceil$ espacio en bloques ocupado por los $CS(A,r)$ registros (los que satisfacen la selección)

- 1, menos el que ya se ha encontrado
Para una distribución uniforme de valores se cumple (no siempre es cierto):

$$CS(A,r) = \frac{n_r}{V(A,r)}$$

A3- Índice primario, búsqueda basada en clave.

$E_{A3} = AA_i + 1$, donde AA_i es el número de niveles del índice

2- Condiciones de comparación

2.1 Selección simple $\sigma_{A \le v}(r)$

Tamaño del resultado

El resultado debe tener en media $\frac{n_r}{2}$ tuplas.

Más preciso (se asume distribución uniforme): $\begin{cases} 0, v < \min(A,r) \\ n_r, v \ge \max(A,r) \\ n_r \dfrac{v - \min(A,r)}{\max(A,r) - \min(A,r)}, e.o.c. \end{cases}$

A6- Índice primario, comparación

Para $A < v$ y $A \le v$ no es necesario buscar en el índice: se itera desde el primero.

Para $A > v$ o $A \ge v$ se busca en el índice el valor $A = v$.

Si la mitad de los registros cumplen la condición:

$$E_{A6} = AA_i + \frac{b_r}{2}$$

Si hay c registros estimados que cumplen la condición:

$$E_{A6} = AA_i + \left\lceil \frac{c}{f_r} \right\rceil$$

2.2 Selección compleja $\sigma_{\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n}(r)$

Tamaño del resultado

$t_i = N(\sigma_{\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n}(r))$, siendo N la función que calcula una estimación del tamaño de la consulta.

$\frac{t_i}{n_r}$ es la probabilidad de que se cumpla θ_i

$\prod_{i=1}^{n} \frac{t_i}{n_r}$ es la probabilidad de que se cumpla $\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n$.

$n_r \prod_{i=1}^{n} \frac{t_i}{n_r} = n_r^{1-n} \prod_{i=1}^{n} t_i$ es el tamaño estimado de la selección compleja.

2.3 Selección compleja $\sigma_{\theta_1 \vee \theta_2 \vee \cdots \vee \theta_n}(r)$

$t_i = N(\sigma_{\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n}(r))$, siendo N la función que calcula una estimación del tamaño de la consulta.

$\frac{t_i}{n_r}$ es la probabilidad de que se cumpla θ_i

La probabilidad de que una tupla cumpla la disyunción es 1 menos la probabilidad de que no cumpla ninguna de las condiciones:

$$1 - (1 - \frac{t_1}{n_r}) * \cdots * (1 - \frac{t_n}{n_r}) = 1 - \prod_{i=1}^{n} (1 - \frac{t_i}{n_r})$$

Algoritmo de ejemplo:

A8- Selección conjuntiva con índice:

Mediante la condición simple más selectiva se leen el menor número de registros (usando los algoritmos A2, A3, ...).
La operación se completa mediante la comprobación en memoria intermedia de que cada registro cumpla el resto de condiciones simples.

8.5.3. Reunión simple (condición θ simple)

Tamaño del resultado

Producto cartesiano $R \times S$, $n_R \cdot n_S$ tuplas = $(n_R \cdot n_S) \cdot (t_R + t_S)$ bytes

Reunión. Casos:

- Si $attrib(R) \cap attrib(S) = \varnothing \Rightarrow R \triangleright\triangleleft S \equiv R \times S$
- Si $attrib(R) \cap attrib(S) =$ una clave de $S \Rightarrow T(R \triangleright\triangleleft S) \leq T(R)$
- Si $attrib(R) \cap attrib(S) =$ clave externa de R y primaria de $S \Rightarrow T(R \triangleright\triangleleft S) = T(R)$
- Si $attrib(R) \cap attrib(S) \neq \varnothing$ no es clave:

  Cada tupla t de R, con $attrib(R) \cap attrib(S) = \{A\}$, produce (estimación) $\dfrac{n_S}{V(A,S)}$, y para todas las tuplas de $R : n_R \dfrac{n_S}{V(A,S)}$.

  Desde el otro punto de vista: $n_S \dfrac{n_R}{V(A,R)}$

  Si $V(A,R) \neq V(A,S)$ son estimaciones distintas, la más baja de ambas es la adecuada.

Algoritmo de bucle anidado:
(todo cabe en memoria intermedia)

for each tupla t_R **in** R **do begin**
    for each tupla t_S **in** S **do begin**
        Comprobar que el par (t_R, t_S) satisface la condición θ de la reunión.
        Si la cumple, añadir $t_R \cdot t_S$ al resultado.
    end
end

Más eficaz con índices.

Algoritmo de bucle anidado por bloques:
(memoria intermedia pequeña)

for each bloque B_R **of** R **do begin**
  for each bloque B_S **of** S **do begin**
    for each tupla t_R **in** B_R **do begin**
      for each tupla t_S **in** B_S **do begin**
        Comprobar que el par (t_R, t_S) satisface la condición de la reunión
        Si la cumple, añadir $t_R \cdot t_S$ al resultado.
            end
        end
    end
end

8.5.4. Reunión compleja (condición θ compuesta)

a) Conjunción $\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n$

$$R \bowtie_{\theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n} S = \sigma_{\theta_1 \wedge \cdots \wedge \theta_{i-1} \wedge \theta_{i+1} \wedge \cdots \wedge \theta_n} (R \bowtie_{\theta_i} S)$$

b) Disyunción $\theta_1 \vee \theta_2 \vee \cdots \vee \theta_n$

$$R \bowtie_{\theta_1 \vee \theta_2 \vee \cdots \vee \theta_n} S = (R \bowtie_{\theta_1} S) \cup (R \bowtie_{\theta_2} S) \cup \cdots \cup (R \bowtie_{\theta_n} S)$$

Se aplican los algoritmos anteriores a σ y $\bowtie_{\theta}$.

También se puede elegir qué reunión hacer primero.

8.5.5. Eliminación de duplicados

a) Ordenar y eliminar consecutivos repetidos
b) Crear una tabla hash y añadir sólo si no está repetido

8.5.6. Proyección

a) Iteración sobre las tuplas generando tuplas sin los atributos necesarios.
b) Eliminación de duplicados si algún atributo de la lista de proyección no es clave

8.5.7. Operaciones sobre conjuntos

Válido para unión, intersección y diferencia

a) Mediante ordenación
   - Ordenar las relaciones
   - Explorar ....

b) Mediante tablas hash
   ....

Bibliografía

[ACPT00] P. Atzeni, S. Ceri, S. Paraboschi y R. Torlone, "Database Systems. Concepts, Languages and Architectures", McGraw-Hill, 2000.

[Dat93] C.J. Date, "Introducción a los Sistemas de Bases de Datos", Addison-Wesley, Reading Massachusetts, 1993.

[EN00] R. Elmasri y S.B. Navathe, "Fundamentals of Data Base Systems", Addison-Wesley, 2000.
Apartados 8.1, 8.2, 8.3, 8.5

[Pres98] Pressman, R.S., "Ingeniería del software. Un enfoque práctico", McGraw-Hill, 1998.
Apartado 8.4

[SKS98] SILBERSCHATZ, A., KORTH, H.F., SUDARSHAN, S. "Fundamentos de Bases de Datos", 3ª edición, McGraw-Hill, 1998.
Apartados Apartados 8.1, 8.2, 8.4

[Ull98] J.D. Ullman, "Principles of Database and Knowledge Base Systems", Vol. I y II, Computer Science Press, 1998.

[Wir94] N. WIRTH, "Algoritmos + estructuras de datos = programas", 1994 (reimpresión).
Apartado 8.4.1