

Universitat de Lleida

Activitat d'Avaluació

Optimització de Consultes

Fet per:

Oriol Alàs Cercós, Àiax Faura Vilalta, Joan Martí Olivart,

Dia de termini:

10 de Març 2020

Universitat de Lleida
Escola Politècnica Superior
Grau en Enginyeria Informàtica
Ampliació de Bases de Dades i Enginyeria del Programari

Professorat:

Juan Manuel Gimeno

Contents

1	Exercici 1	1
1.1	Càlcul del volum de dades	2
2	Exercici 2	4
2.1	Estratègia A	4
2.1.1	Selecció de la relació alquiler	4
2.1.2	Guardar la selecció	4
2.1.3	Join vehicle - selecció alquiler	5
2.1.4	Cost total	5
2.2	Estratègia B	5
2.2.1	Selecció de la relació vehicle	5
2.2.2	Selecció de la relació alquiler	6
2.2.3	Guardar les seleccions	6
2.2.4	Join de les dues seleccions	6

Índex de figures

1	Proposta d'arbre equivalent a la consulta	1
2	Arbre un cop baixades les projeccions.	1
3	Arbre sense les projeccions redundants.	2
4	Arbre amb els diferents passos que segueix la selecció.	2

1 Exercici 1

Proposa un arbre equivalent a la següent consulta, calcula el volum de dades que es va tractant al llarg de tot l'arbre i fes la seva optimització sintàctica.

L'arbre dissenyat equivalent a la consulta es pot veure a la figura 1.

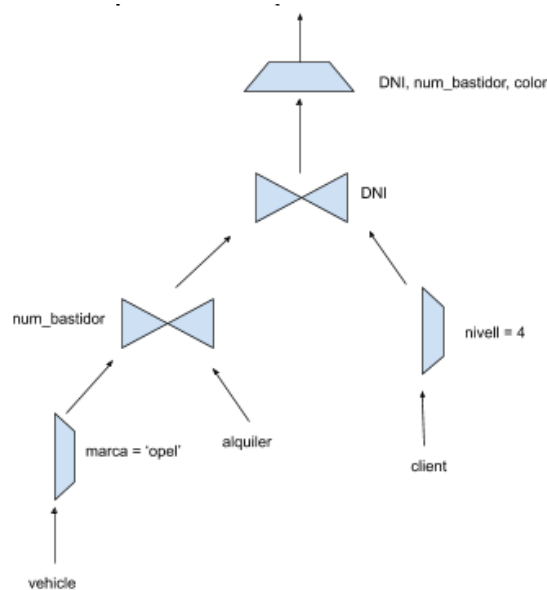


Figure 1: Proposta d'arbre equivalent a la consulta

Un cop dissenyat, es realitzen els diferents passos a seguir.

1. *Baixar les seleccions.* Com es pot observar a la figura 1, les seleccions estan en el nivell més baix possible.
2. *Reagrupar seleccions.* No hi ha seleccions a agrupar.

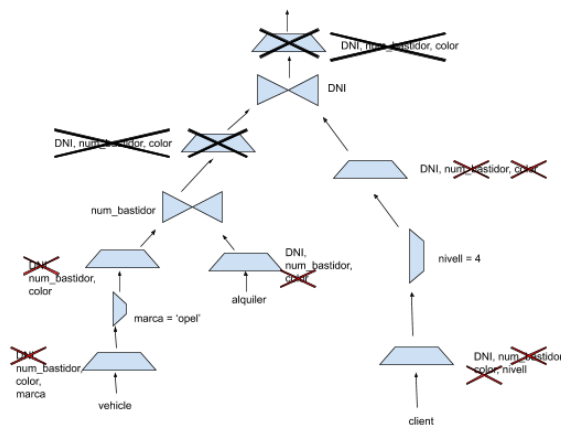


Figure 2: Arbre un cop baixades les projeccions.

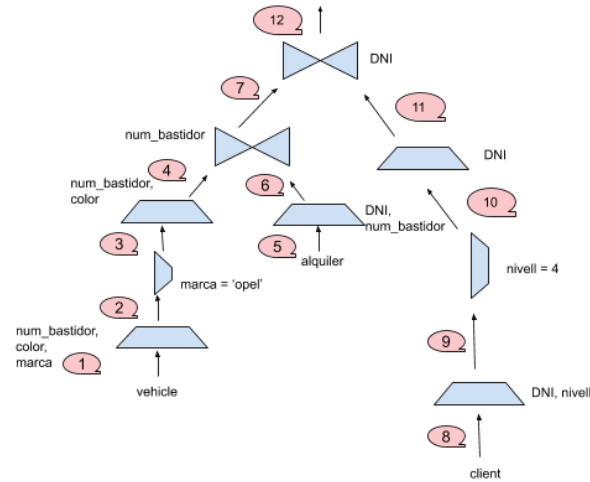


Figure 3: Arbre sense les projeccions redundants.

3. *Baixar projeccions.* Un cop baixades les projeccions, hem tret tots aquells atributs que no es troben, tal i com s'observa a la figura 2.
4. El resultat un cop simplificant les projeccions, treient aquelles que no realitzen cap tipus de canvi es pot veure a la figura 3.

1.1 Càlcul del volum de dades

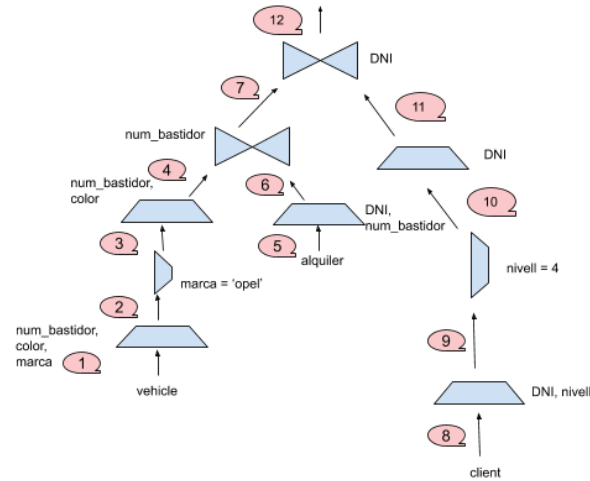


Figure 4: Arbre amb els diferents passos que segueix la selecció.

Per indicar que la cardinalitat d'un pas fa referència a un pas anterior, s'ha indicat el nombre del pas al que fa referència subrantant-lo. Exemple: $\text{card}(6) = \text{cardinalitat que es tenia en el pas 6}$.

1. $\text{card}(\text{vehicle}) = 50$
 $\text{mida}(\text{vehicle}) = 3 + 15 + 15 + 5 = 38$
 $\text{volum} = 50 \times 38 = 1900$

2. $\text{card}(1) = 50$
 $\text{mida} = 3 + 15 + 5 = 23$
 $\text{volum} = 50 \times 23 = 1150$
 3. $\text{card}(2) = 50 \times 0,2 = 10$
 $\text{mida} = 23$
 $\text{volum} = 10 \times 23 = 230$
 4. $\text{card}(3) = 10$
 $\text{mida} = 3 + 5 = 8$
 $\text{volum} = 10 \times 8 = 80$
 5. $\text{card}(\text{alquiler}) = 50$
 $\text{mida}(\text{alquiler}) = 3 + 3 + 6 = 12$
 $\text{volum} = 50 \times 12 = 600$
 6. $\text{card}(5) = 50$
 $\text{mida} = 3 + 3 = 6$
 $\text{volum} = 50 \times 6 = 300$
 7. $\text{card}(\max(4,6)) = \text{card}(6) = 50$
 $\text{mida} = 8 + 6 - 3 = 11$
 $\text{volum} = 50 \times 11 = 550$
 8. $\text{card}(\text{client}) = 250$
 $\text{mida} = 3 + 30 + 15 + 3 = 51$
 $\text{volum} = 250 \times 51 = 12750$
 9. $\text{card}(8) = 250$
 $\text{mida} = 3 + 3 = 6$
 $\text{volum} = 250 \times 6 = 1500$
 10. $\text{card}(9) = 250 \times 0.25$
 $\text{mida} = 6$
 $\text{volum} = 63 \times 6 = 378$
 11. $\text{card}(10) = 63$
 $\text{mida} = 3$
 $\text{volum} = 63 \times 3 = 189$
 12. $\text{card}(\max(7,11)) = \text{card}(11) = 63$
 $\text{mida} = 11 + 3 - 3 = 11$
 $\text{volum} = 63 \times 11 = 693$
- Total = $1900 + 1150 + 230 + 80 + 600 + 300 + 550 + 12750 + 1500 + 378 + 189 + 693$
Total = 20320

2 Exercici 2

Describeu dues estratègies d'execució per a resoldre la següent consulta, tot indicant les possibles combinacions d'algorismes d'implementació de cada operació segons que cada algorisme, per a cada operació, sigui o no aplicable.

Cal deixar indicada, mitjançant la fórmula i les dades corresponents, l'estimació de costos en les opcions aplicables (no poseu simplement el resultat final) i justificar (1 línia) l'aplicabilitat o no de l'algorisme.

2.1 Estratègia A

Selecció de la relació alquiler i després fer el join d'alquiler i vehicle i després seleccionar la relació segons l'atribut marca provinent de vehicle.

2.1.1 Selecció de la relació alquiler

- 1r Algorisme: No realitzable ja que fecha-devolució no és índex cluster
- 2n Algorisme: No realitzable ja que fecha-devolució no és índex cluster
- 3r Algorisme: No realitzable ja que fecha-devolució no és índex no cluster
- 4t Algorisme: Lectura seqüencial.

$$cost = B_{alquiler} = 3$$

- 5è Algorisme: No realitzable ja que fecha-devolució no és índex no cluster
- 6è Algorisme: No realitzable perquè la relació no està en diferents fitxers.
- 7è Algorisme; No realitzable perquè fecha-devolució no és índex cluster.

Per tant, l'algorisme òptim per realitzar la consulta és el 4t.

2.1.2 Guardar la selecció

Es busca la cardinalitat de la selecció:

$$card(seleccio) = card(alquiler) * percentatge_{fecha-devolucion > '21-2-2020'} = 50 * \frac{35}{100} = 17.5 \simeq 18$$

Per tal de trobar el nombre de blocs de la selecció, s'ha de trobar el factor de bloqueig, que és:

$$b_{alquiler} = \frac{mida_{pagina}}{mida_{tupla}} = \frac{50}{3} = 16.6 \simeq 17$$

$$B_{alquiler} = \frac{card(seleccio)}{b_{alquiler}} = \frac{17.5}{17} \simeq 2$$

2.1.3 Join vehicle - selecció alquiler

- **Bucles imbricats.** Com que es vol que el bucle amb menys iteracions sigui l'exterior:

$$cost = B_{seleccio} + B_{seleccio} * B_{vehicle} = 2 + 2 * 10 = 22$$

- **Ordenació-fusió.** La relació vehicle ja està ordenada per *numbastidor*, doncs aquesta és la seva clau primària. Llavors, només s'ha de tenir en compte que la selecció no està ordenada per aquest atribut.

$$\begin{aligned} cost &= 2 * \log_2(B_{seleccio}) * B_{seleccio} + 2 * \log_2(B_{vehicle}) * B_{vehicle} + B_{seleccio} + B_{vehicle} = \\ &= 2 * \log_2(2) * 2 + 2 * \log_2(10) * 10 + 2 + 10 = 16 \end{aligned}$$

- **Bucle amb índex.** Es pot aplicar perquè vehicle té l'índex cluster num-bastidor i està ordenat per aquest índex. Com a màxim, hi haurà 18 tuples que facin join, per tant, el cost d'aquest algorisme és:

$$cost = B_{seleccio} + card(seleccio) * S * card(seleccio) * card(vehicle) = 2 + 17.5 * 2 * 18 = 37$$

Per tant, l'algorisme per realitzar el join és el d'ordenació-fusió.

2.1.4 Cost total

$$cost_{EstrategiaA} = 3 + 2 + 16 = 21$$

2.2 Estratègia B

La segona estratègia seleccionada ha estat fer primer les seleccions i després la combinació (join entre ambdòs seleccions).

2.2.1 Selecció de la relació vehicle

- 1r Algorisme: No realitzable ja que marca no és un índex cluster.
- 2n Algorisme: No realitzable ja que marca no és un índex cluster.
- 3r Algorisme. Realitzable ja que marca és un índex cluster i la selecció és de la forma

$$A_i = cte$$

$$cost = S(marca = 'opel') * card(vehicle) = \frac{card(vehicle)}{NDIST(marca)} = \frac{50}{7} = 8$$

- 4t Algorisme. Lectura seqüencial.

$$cost = B_{vehiculo} = 10$$

- 5è Algorisme. No realitzable ja que la selecció no és de la forma: $A_i \geq, \leq, <, >$
- 6è Algorisme. No realitzable ja que la relació no està en diferents fitxers.
- 7è Algorisme. Combinació d'indexos. És igual que l'algorisme 3, ja que no més hi ha un índex de vehicle que intervingui amb la consulta.

Per tant, l'algorisme òptim per realitzar la consulta és el 3r.

2.2.2 Selecció de la relació alquiler

La selecció de la relació alquiler és el mateix que en l'estratègia A, així que l'algorisme òptim per realitzar la consulta és el 4t amb $cost = 3$

2.2.3 Guardar les seleccions

Per guardar les seleccions, hem de saber les cardinalitats de les consultes.

$$card(query_{vehiculo} = card(vehiculo) * percentatge_{marca='Opel'} = 50 * \frac{20}{100} = 10$$

$$B_{vehiculo} = \frac{card(vehiculo)}{b_{vehiculo}} \Rightarrow b_{vehiculo} = \frac{card(vehiculo)}{B_{vehiculo}} = \frac{50}{10} = 5$$

$$B_{query_{vehiculo}} = \frac{card(query_{vehiculo})}{b_{vehiculo}} = \frac{10}{5} = 2$$

$$card(query_{alquiler} = card(alquiler) * percentatge_{fecha-devolucion > '21-2-2020'} = 50 * \frac{35}{100} = 17.5 \simeq 18$$

Per tant, el cost de guarda la query de vehicle és 2.

$$B_{alquiler} = \frac{card(alquiler)}{b_{alquiler}} \Rightarrow b_{alquiler} = \frac{card(alquiler)}{B_{alquiler}} = \frac{50}{3} = 17$$

$$B_{query_{vehiculo}} = \frac{card(query_{vehiculo})}{b_{vehiculo}} = \frac{18}{17} = 2$$

Per tant, el cost de guarda la query de vehiculo és 2.

$$cost_{total} = 2 + 2 = 4$$

El cost total de guardar les dues seleccions és 4.

2.2.4 Join de les dues seleccions

- **Bucles imbricats.** Com que qualsevol de les dues tenen el mateix nombre de blocs, no importa quin és el bucle exterior o interior.

$$cost = B_R + B_R * B_S = 2 + 2 * 2 = 6$$

- **Ordenació-fusió.** Com que cap de les dues estan ordenades, s'han de realitzar les ordenacions.

$$\begin{aligned} cost &= 2 * \log_2(B_R) * B_R + 2 * \log_2(B_S) * B_S + B_R + B_S = \\ &= 2 * \log_2(2) * 2 + 2 * \log_2(2) * 2 + 2 + 2 = 12 \end{aligned}$$

- **Bucle amb índex.** No realitzable perquè al realitzar les dues seleccions ja no es poden utilitzar els índexs.

Per tant, l'algorisme per realitzar el join és el de bucles imbricats, amb un cost de 6.

2.2.5 Cost total

$$cost_{EstrategiaB} = 8 + 3 + 4 + 6 = 21$$