Parcialito 5 - Optimización de Consultas - CORRECCIÓN RÁPIDA

Graded

Student

Juana Rehl

Total Points

10 / 10 pts

Question 1

Estime el costo de realizar la consulta y la cantidad de filas que serán devueltas.

10 / 10 pts

✔ **– 0 pts** Correcto

– 1 pt Mal calculada la cardinalidad por fechas

– 1 pt Error de cálculos

– 2.5 pts Falta calcular cantidad de filas de la salida (cardinalidad)

– 2 pts Error conceptual

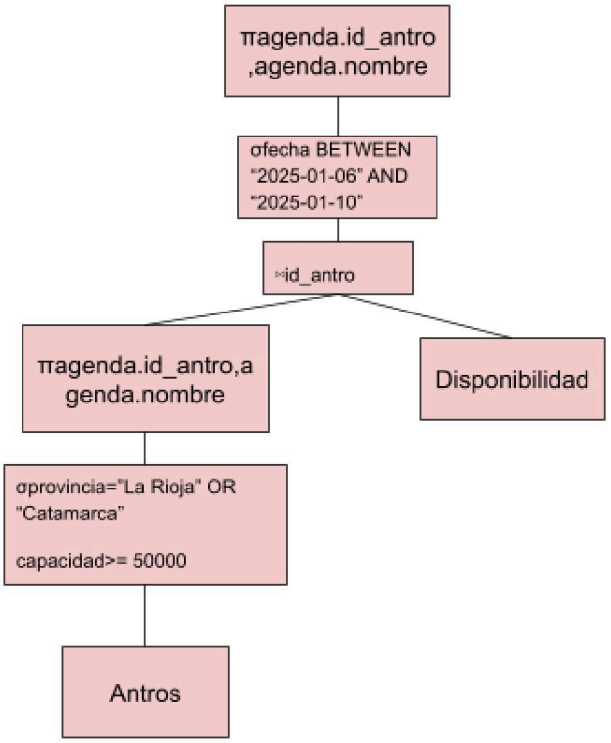
– 2 pts Falta probar casos de join

1. Por qué no aparecería B(R) o B(R')?
2. Aparte no tenés índice para hacer la junta después

Question assigned to the following page: 1

Juana Rehl - 112185 - Parcialito 5
Resolución

R = Antros
S = Disponibilidad



1. Aplicación de selección sobre R. Primero se filtra la relación Antros por provincia y capacidad:

R' = σ((provincia = 'La Rioja' OR provincia = 'Catamarca') ∧ (capacidad >= 50000))(R) ✔️

Costo por capacidad:

- FileScan → `B(R) = 2000` bloques, (No hay índice sobre capacidad).

Costo por provincia:

- IndexScan → `H(I(A,R)) + ⌜n(R)/V(A,R)⌝ = 2 + 10000/20 = 502`.
  Como se trata de dos valores (por el OR), se duplica: `502 × 2 = 1004`. ✔️

Se elige el método por provincia, ya que resulta ser el de menor costo en comparación al de capacidad. COST(σprovincia,capacidad(R)) = 1004.

Question assigned to the following page: 1

Cantidad de filas filtradas:
`n(R') = (10000 × 2 / 20) × 0.1 = 100`

Para poder sacar el n(R') se tiene en cuenta que de las 20 provincias se precisan 2 de ellas y tenemos información de que el 10% de los antros tienen capacidad para al menos 50 mil personas (por eso lo multiplicamos por 0.1).
Factor de bloque: `F(R) = 10000 / 2000 = 5`
Cantidad de bloques: `B(R') = 100 / 5 = 20` ✔️

En este punto, se podría plantear otra alternativa e ir por otro enfoque. Por ejemplo en vez de seleccionar por provincias/capacidad, seleccionar según la fecha (σfecha(s)). Aunque es una alternativa completamente válida, rápidamente nos podemos dar cuenta que no es la más óptima.
Si realizamos un FileScan por fecha: FileScan →5000. Por IndexScan → 3 + 100000 * 5/100 = 5003. Entre estas dos opciones nos conviene el FileScan, pero si lo comparamos con el enfoque seleccionado anteriormente, nos damos cuenta que el costo de seleccionar por provincias/capacidad es de 1004 que es mucho menor que 5000. Descartamos esta alternativa por fechas. ✔️

2. Evaluación del JOIN (R' ⋈ S). Disponemos de M = 100 bloques de memoria.

a) Loops anidados por bloque:
`COST = ⌈B(R')/(M-2)⌉ × B(S) = 1 × 5000 = 5000` ✔️

b) Loop con único índice:
`COST = n(R') × (H(I(A,S)) + ⌈n(R')/V(A,S)⌉) = 100 × (4 + 100000/10000) = 1400` ✔️

c) Sort-Merge:
`COST = B(S) + 2×B(R')×⌈logM-1(B(R'))⌉ + 2×B(S)×⌈logM-1(B(S))⌉ = 10000 + 2×20×1 + 2×10000×1 = 30040` ✔️

d) Hash Join:

Límites: P <= M - 1 <=99 y min(1; 51) <= M - 2 <= 98
`COST = 2×B(R') + 3×B(S) = 2×20 + 3×5000 = 15040` ✔️

El método de loop con único índice resulta el más eficiente, con costo total = 1400.

Cantidad de filas luego del join:
`n(R'⋈S) = 100 × (100000 / 10000) = 1000` ✔️

Factor de bloque:
`F(R'⋈S) = 5×20 / (5 + 20) = 4`
Cantidad de bloques: `B(R'⋈S) = 1000 / 4 = 250` ✔️

Question assigned to the following page: 1

3. Filtro por fecha. Sea Y = R'⋈S. Aplicamos el filtro sobre `fecha` (entre el 6 y el 10 de enero). El operador se ejecuta en pipeline, por lo que no incurre en costo adicional de lectura. COST(σfecha(Y)) = 0 ✔️

4. Proyección final

Como no se requiere eliminar duplicados, la proyección se realiza directamente sobre la salida del operador anterior, sin costo extra. ✔️

Costo total estimado

`Ctotal = 1004 (selección) + 1400 (join) = 2404`

Filas estimadas

`n(Y) = 1000 × 0.05 = 50` (ya que se toman 5 de las 100 fechas posibles) ✔️

El plan más eficiente consiste en usar el join con índice único, con un costo total aproximado de 2404 bloques y una salida esperada de 50 registros.