Sistemas de Bases de Datos 2

2024

Ing. Luis Alberto Arias Solórzano

Unidad 3

Estimación

- La estimación del tamaño del resultado de una operación de selección depende del predicado de la selección.

- En primer lugar se considerará un solo predicado de igualdad, luego un solo predicado de comparación y, finalmente, combinaciones de predicados.

- $\sigma A = a (r)$: Si se supone una distribución uniforme de los valores (es decir, que cada valor aparece con igual probabilidad), se puede estimar que el resultado de la selección tiene $nr/V(A, r)$ tuplas, suponiendo que el valor a aparece en el atributo A de algún registro de r. La suposición de que el valor

- En algunos casos, como cuando la consulta forma parte de un procedimiento almacenado, puede que el valor v no esté disponible cuando se optimice la consulta. En esos casos, se supondrá que aproximadamente la mitad de los registros cumplen la condición de comparación. Es decir, se supone que el resultado tiene $nr/2$ tuplas; la estimación puede resultar muy imprecisa, pero es lo mejor que se puede hacer sin más información.

Estimación

- **Selecciones complejas:**

- – **Conjunción:** Una *selección conjuntiva* es una selección de la forma

  $\sigma\theta_1 @ \theta_2 @ \cdots @ \theta_n (r)$ Se puede estimar el tamaño del resultado de esta selección: Para cada θ_i, se estima el tamaño de la selección $\sigma\theta_i (r)$, denotada por s_i, como se ha descrito anteriormente.

- Por tanto, la probabilidad de que una tupla de la relación satisfaga la condición de selección θ_i es s_i/n_r. La probabilidad anterior se denomina **selectividad** de la selección $\sigma\theta_i (r)$. Suponiendo que las condiciones sean *independientes* entre sí, la probabilidad de que una tupla satisfaga todas las condiciones es simplemente el producto de todas las combinaciones posibles.

Estimación

- **Disyunción:** Una *selección disyuntiva* es una selección de la forma σθ1 2 θ2 2 … 2θ*n* (*r*) Una condición disyuntiva se satisface por la unión de todos los registros que satisfacen las condiciones individuales y simples θ*i*. Como anteriormente, admitamos que *si /nr* denota la probabilidad de que una tupla satisfaga la condición θ*i*. La probabilidad de que la tupla satisfaga la disyunción es, pues, 1 menos la probabilidad de que no satisfaga *ninguna* delas condiciones:

- Multiplicando este valor por *n r* se obtiene el número estimado de tuplas que satisfacen la selección.

Estimación

- **Negación:** En ausencia de valores nulos el resultado de una selección σθ (*r*) es simplemente las tuplas de *r* que no están en σθ (*r*). Ya se sabe el modo de estimar el número de tuplas de σθ (*r*). El número de tuplas de σθ (*r*) se estima, por lo tanto, que es *n* (*r*) menos el número estimado de tuplas de σθ (*r*).

- Se pueden tener en cuenta los valores nulos estimando el número de tuplas para las que la condición θ se evalúa como *desconocida*, y restar ese número de la estimación anterior que ignora los valores nulos. La estimación de ese número exige conservar estadísticas adicionales en el catálogo.

Estimación

Estimación del tamaño de las reuniones

- El producto cartesiano $r \times s$ contiene $nr * ns$ tuplas. Cada tupla de $r \times s$ ocupa $tr + ts$ bytes, de donde se puede calcular el tamaño del producto cartesiano.

- La estimación del tamaño de una reunión natural resulta algo más complicada que la estimación del tamaño de una selección del producto cartesiano. Sean r (R) y s (S) dos relaciones. • Si $R \cap S = \emptyset$ —es decir, las relaciones no tienen ningún atributo en común— entonces $r\,s$ es igual que $r \cap s$, y se puede utilizar la técnica de estimación anterior para los productos cartesianos.

- • Si $R \cap S$ es una clave de R, entonces se sabe que cada tupla de s se combinará como máximo con

- El caso más difícil es que $R \cap S$ no sea una clave de R ni de S. En este caso se supone, como se hizo para las selecciones, que todos los valores aparecen con igual probabilidad. Considérese una tupla t de r y supóngase que R

Estimación

Estimación del tamaño de otras operaciones

- **Proyección:** El tamaño estimado (número de registros de las tuplas) de una proyección de la forma $\Pi A(r)$ es $V(A, r)$, ya que la proyección elimina los duplicados.

- **Agregación:** El tamaño de $A\ G\ F(r)$ es simplemente $V(A, r)$, ya que hay una tupla de $A\ G\ F(r)$ por cada valor distinto de A.

- **Operaciones de conjuntos:** Si las dos entradas de una operación de conjuntos son selecciones de la misma relación se puede reescribir la operación de conjuntos como disyunciones, conjunciones o negaciones.

Por ejemplo, $\sigma\theta_1(r) \cup \sigma\theta_2(r)$ puede reescribirse como $\sigma\theta_1 2\theta_2(r)$. De manera parecida, se pueden reescribirlas intersecciones como conjunciones y la diferencia de conjuntos empleando la negación, siempre que las dos relaciones que participan en la operación de conjuntos sean selecciones de la misma relación.

- **Reunión externa:** El tamaño estimado de $r\ s$ es el tamaño de $r\ s$ más el tamaño de r; el de $r\ s$ es simétrico, mientras que el de $r\ s$ es el tamaño de $r\ s$ más los tamaños de r y de s. Las tres estimaciones pueden ser imprecisas, pero proporcionan cotas superiores para los tamaños.

Optimización basada en el coste

- Los **optimizadores basados en el coste** generan una gama de planes de evaluación a partir de la consulta dada empleando las reglas de equivalencia y escogen el de coste mínimo. Para las consultas complejas el número de planes de consulta diferentes que son equivalentes a un plan dado puede ser grande.

- El procedimiento almacena los planes de evaluación que calcula en el array asociado *mejor plan*, que está indexado por conjuntos de relaciones. Cada elemento del array asociativo contiene dos componentes: el coste del mejor plan de *S* y el propio plan. El valor de *mejor plan*[*S*].*coste* se supone que se inicializa como ∞ si *mejor plan*[*S*] no se ha calculado todavía. El procedimiento comprueba en primer lugar si el mejor plan para calcular la reunión del conjunto de relaciones dado *S* se ha calculado ya (y se ha almacenado en el array asociativa *mejor plan*); si es así, devuelve el plan ya calculado. Si *S* sólo contiene una relación, se calcula la mejor forma de acceder a *S* (teniendo en cuenta las relaciones sobre *S*, si las hay) y se almacena en *mejor plan*.

- En caso contrario, el procedimiento intenta todas las maneras posibles de dividir *S* en dos subconjuntos disjuntos. Para cada división el procedimiento halla de manera recursiva los mejores planes para cada uno de los dos subconjuntos y luego calcula el coste del plan global utilizando esa división. El procedimiento escoge el plan más económico de entre todas las alternativas para dividir *S* en dos conjuntos. El procedimiento almacena el plan más económico y su coste en el array *mejor plan* y los devuelve.

Optimización Heurística

- Un inconveniente de la optimización basada en el coste es el coste de la propia optimización. Aunque el coste del procesamiento de las consultas puede reducirse mediante optimizaciones inteligentes, la optimización basada en el coste sigue resultando costosa. Por ello, muchos sistemas utilizan la **heurística** para reducir el número de elecciones que hay que hacer de una manera basada en los costes. Algunos sistemas incluso deciden utilizar sólo la heurística y no utilizan en absoluto la optimización basada en el coste. Un ejemplo de regla heurística es la siguiente regla para la transformación de consultas del álgebra relacional:

• Llevar a cabo las operaciones de selección tan pronto como sea posible. Los optimizadores heurísticos utilizan esta regla sin averiguar si se reduce el coste mediante esta transformación.

Optimización Heurística

- La operación de proyección, como la operación de selección, reduce el tamaño de las relaciones. Por tanto, siempre que haya que generar una relación temporal, resulta ventajoso aplicar inmediatamente cuantas proyecciones sea posible. Esta ventaja sugiere un acompañante a la heurística «llevar a cabo las selecciones tan pronto como sea posible»:

• Llevar a cabo las proyecciones tan pronto como sea posible.

Optimización Heurística

- Hay que descomponer las selecciones conjuntivas en una secuencia de operaciones de selección sencillas. Este paso, basado en la regla de equivalencia:

1. Facilita el desplazamiento de las operaciones de selección hacia la parte inferior del árbol de consultas.

2. Hay que desplazar las operaciones de selección hacia la parte inferior del árbol de consultas para conseguir su ejecución lo antes posible. Este paso utiliza las propiedades de conmutatividad y de distributividad de la operación de selección puestas de manifiesto en las reglas de equivalencia 2,7.a, 7.b y 11.

Optimización Heurística

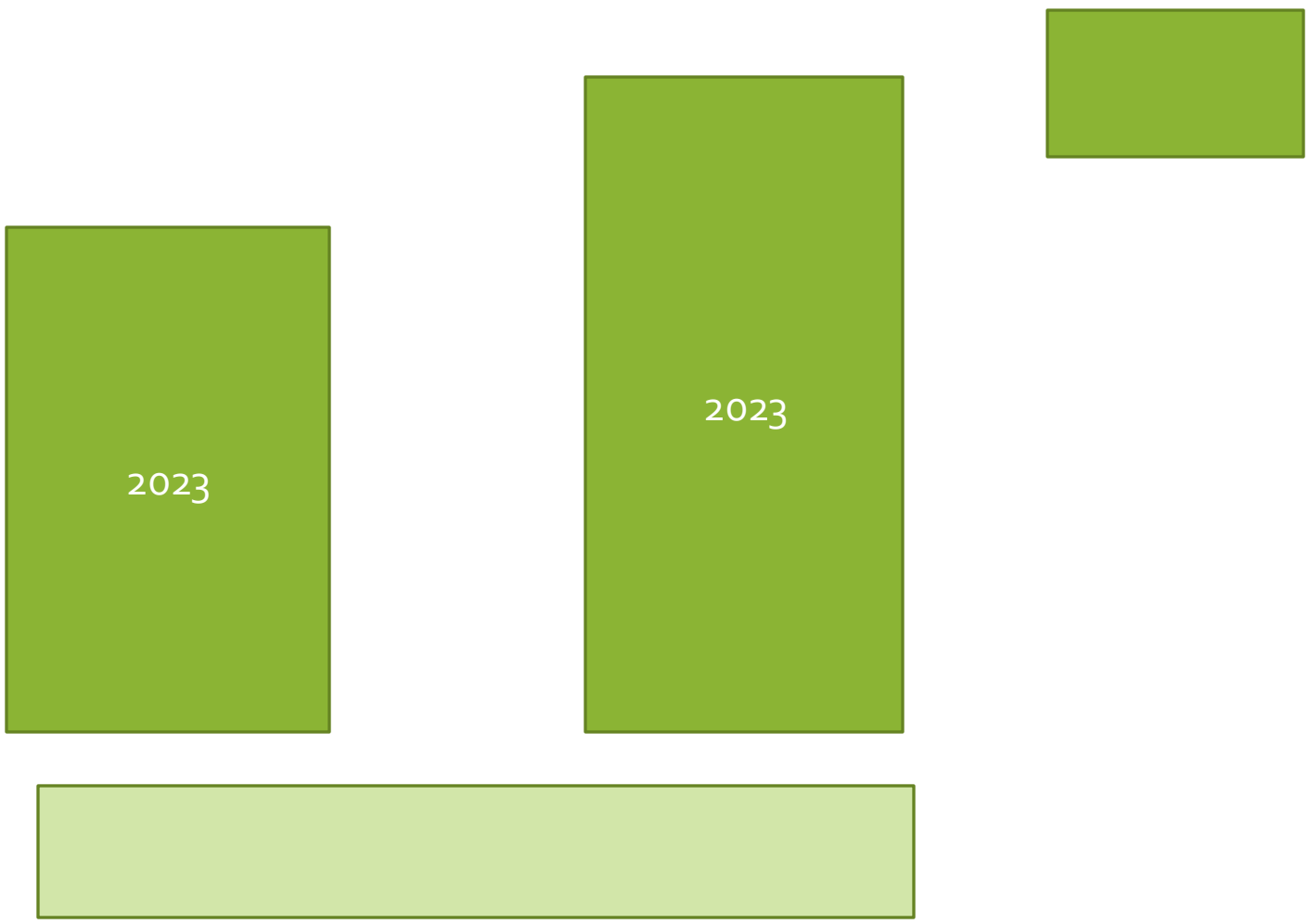
- **3.** Hay que determinar las operaciones de selección y de reunión que producirán las relaciones de menor tamaño, es decir, que producirán las relaciones con el menor número de tuplas. Utilizando la asociatividad de la operación hay que reordenar el árbol para que las relaciones de los nodos hojas con esas selecciones restrictivas se ejecuten antes. Este paso considera la selectividad de las condiciones de selección o de reunión. Hay que recordar que la selección más restrictiva —es decir, la condición con la selectividad de menor tamaño— recupera el menor número de registros. Este paso confía en la asociatividad de las operaciones binarias dada en la regla de equivalencia

- **4.** Hay que sustituir por operaciones de reunión las operaciones producto cartesiano seguidas de condiciones de selección (regla 4.a). La operación producto cartesiano suele resultar costosa de implementar, ya que $r1 \times r2$ incluye un registro por cada combinación de registros procedentes de $r1$ y de $r2$. La selección puede reducir de manera significativa el número de registros, haciendo la reunión mucho menos costosa que el producto cartesiano.

Optimización Heurística

5. Hay que dividir las listas de atributos de proyección y desplazarlas hacia la parte inferior del árbol todo lo que sea posible, creando proyecciones nuevas donde sea necesario. Este paso se aprovecha de las propiedades de la operación de proyección dadas en las reglas de equivalencia.

6. Hay que identificar los subárboles cuyas operaciones pueden encauzarse y ejecutarlos utilizando el encauzamiento.


2023
2023

Gracias