2.10

El factor de selectividad de una operación, es decir, la proporción de tuplas de una relación de operando que participan en el resultado de esa operación, se denota como SFOP, donde OP denota la operación.

La cardinalidad de la selección es

card**(s**F**(** R**)) =**  SFS**(** F**)** \_ card**(** R**)**

donde SFS (F) depende de la fórmula de selección y se puede calcular de la siguiente manera [Selinger et al., 1979], donde p (Ai) yp (Aj) indican predicados sobre los atributos Ai y Aj, respectivamente:

SFS**(** A **=**  value**) =**

1

card**(P**A**(** R**))**

SFS**(** A > value**) =**

max**(** A**)**􀀀 value

max**(** A**)**􀀀 min**(** A**)**

SFS**(** A < value**) =**

value􀀀 min**(** A**)**

max**(** A**)**􀀀 min**(** A**)**

SFS**(** p**(** Ai**)**^ p**(** Aj**)) =**  SFS**(** p**(** Ai**))** \_ SFS**(** p**(** Aj**))**

SFS**(** p**(** Ai**)**\_ p**(** Aj**)) =**  SFS**(** p**(** Ai**))+** SFS**(** p**(** Aj**))**􀀀**(** SFS**(** p**(** Ai**))** \_ SFS**(** p**(** Aj**)))**

SFS**(** A 2 f valuesg**) =**  SFS**(** A **=**  value**)** \_ card**(**f valuesg**)**

Pag 253

2.11

Con la optimización de consultas estáticas, existe una separación clara entre la generación del QEP en tiempo de compilación y su ejecución por el motor de ejecución de DBMS. Por lo tanto, un modelo de costo preciso es clave para predecir los costos de los candidatos QEPs

Pag 261

2.12

La optimización dinámica de consultas combina las dos fases de descomposición y optimización de consultas con ejecución. El QEP está construido dinámicamente por el optimizador de consultas que realiza llamadas al motor de ejecución de DBMS para ejecutar las operaciones de la consulta. Por lo tanto, no hay necesidad de un modelo de costos.

Pag 257

2.13

La optimización dinámica y estática de las consultas tiene ventajas y desventajas.

La optimización dinámica de consultas mezcla la optimización y la ejecución y así puede hacer elecciones de optimización precisas en tiempo de ejecución. Sin embargo, la optimización de la consulta se repite para cada ejecución de la consulta. Por lo tanto, este enfoque es mejor para consultas ad-hoc.

La optimización de consultas estáticas, realizada en tiempo de compilación, amortiza el costo de la optimización en múltiples ejecuciones de consultas. La precisión del modelo de costos es, por lo tanto, fundamental para predecir los costos de los candidatos QEP. Este enfoque es el mejor para las consultas integradas en procedimientos almacenados, y ha sido adoptado por todos los DBMS comerciales.

Pag 257

2.14

|  |  |  |
| --- | --- | --- |
| Tipos de Algoritmos | Algoritmo | Descripción |
| Dinamica | Ingres | El algoritmo separa recursivamente una consulta expresada en cálculo relacional  (es decir, SQL) en piezas más pequeñas que se ejecutan en el camino. La consulta se descompone primero en una secuencia de consultas que tienen una relación única en común. Luego, cada consulta de monorelación se procesa seleccionando, en función del predicado, el mejor método de acceso a esa relación (por ejemplo, índice, exploración secuencial).Pag 258 |
| Estatica | System R | La entrada al optimizador es un árbol de álgebra relacional resultante de la descomposición de una consulta SQL. El resultado es un QEP que implementa el árbol de álgebra relacional "óptimo".  El optimizador asigna un costo (en términos de tiempo) a cada árbol candidato y retiene el que tiene el menor costo. Los árboles candidatos se obtienen mediante una permutación de las órdenes de unión de las n relaciones de la consulta utilizando las reglas de conmutatividad y asociatividad.Pag 261 |
| Hibrida | Combinación de ambas | La optimización de consultas híbridas intenta proporcionar las ventajas de la optimización de consultas estáticas a la vez que evita los problemas generados por las estimaciones inexactas. El enfoque es básicamente estático, pero pueden tomarse más decisiones de optimización en tiempo de ejecución.  Este enfoque se estableció en el Sistema R al agregar una fase de reoptimización de tiempo de ejecución condicional para los planes de ejecución optimizados estadísticamente [Chamberlin et al., 1981]. Pag 265 |