**Trabajo de Bases de Datos**

Daniel Santa Rendón - C.c. 1040050021

Daniel Torres González - C.c. 1020492000

**Optimización de Consultas (15%)**

**Facultad de Ingeniería**

**Universidad de Antioquia**

Apoyados en la Base de Datos que cada equipo ha diseñado a lo largo del curso resuelva los siguientes interrogantes.

1. **Optimización por costos**
2. **System R. (20%)**

Utilizando el algoritmo de optimización de System-R encuentre la estrategia de menor costo para resolver una consulta de la forma:

***Select***  *col1, col2, …., coln*

***From*** *tabla-r*

***Where*** *col-*i operador-1 *valor-1 AND col-*j operador-2 *valor-2 …;*

*Debe Especificar* los datos de entrada para el algoritmo; es decir:

BR, TR, que índices tiene la tabla , el esquema de almacenamiento usado por la tabla, las imágenes de los atributos con índices, etc.

La consulta debe tener por lo menos tres condiciones en el WHERE, algunas con desigualdad y otras con igualdad.

Que debe entregar:

La Evaluación de cada una de las estrategias posibles para la consulta, indicando su costo.

Indicar cual es la estrategia seleccionada por el optimizador.

**2. Join entre dos tablas (50%)**

Consulta: Debe definir el join desde las tablas de su modelo que siga este patrón.

***SELECT***  *tabla-1.col-1, … , tabla-1.coln,….,tabla-2.col-1,…, tabla-2.colm*

***FROM*** *tabla-1* **JOIN** *tabla-2*  **ON** *tabla-1.col-i = tabla-2.col-j;dfas*

*Premisas :* Hay una clave foránea de tabla-1 hacia tabla-2 con los atributos

usados para construir la condición del join.

El select despliega todos los atributos de ambas tablas.

Ninguna de las dos relaciones se ajusta en memoria principal. Debe especificar M. y recuerde que M < Btabla-1 , M < Btabla-1

Ninguna de las dos relaciones está clasificada (ordenada) de antemano.

*Evaluación:*

**ESCENARIO 1.**

Ninguna de las dos tablas tiene índices en las columnas que hacen join.

Caso 1: Evalúe el join cuando la tabla con más registros tenga menos de 1.000.000 registros, pero más de 500.000. Este decir 500.000 < Tr < 1.000.000

Caso 2: Evalúe el join cuando la tabla con más registros tenga más de 5.000.000 de tuplas.

Es decir, 5.000.000 < Tr < 8.000.000

Tenga presente que los datos de ambas tablas las deben crecer proporcionalmente, según los requerimientos de la aplicación.

Caso 3: Evalúe el join cuando la tabla con más registros tenga más de 10.000.000 de tuplas.

10.000.000 < Tr < inf.

Tenga presente que los datos de ambas tablas las deben crecer proporcionalmente, según los requerimientos de la aplicación.

El valor de M permanece constante para los tres casos.

Haga un cuadro resumen de las estrategias para cada caso e indique cuál estrategia debería utilizar el optimizador en cada uno de ellos.

Debe indicar cuál Relación asume como R y cuál relación asume como S.

Caso X:

|  |  |  |  |
| --- | --- | --- | --- |
| **Datos de entrada por tabla** | **Estrategia Usada** | **Fórmula usada para costo de la entrada** | **Costo entrada.** |
| Esquema de almacenamiento usado por tabla,  Br= Bs=  Tr= Ts=  Imágenes de atributo con índices.  Ir.b =  Ir.s =  M= | Selección en un producto |  |  |
| Sort Merge |  |  |
| Hash Join. |  |  |

NOTAS:

Si sus tablas tienen menos registros de los indicados asuma los valores que en este punto se establecen. Recuerde que la base de datos debe crecer entonces en todas sus tablas de una manera proporcional.

**ESCENARIO 2.**

Las tablas tiene índice en los atributos del join. Es decir *tabla-1.col-i y tabla-2.col-j tienen indices.*

Caso 1: Evalúe el join cuando la tabla con más registros tenga más de 5.000.000 de tuplas.

Es decir, 5.000.000 < Tr < 8.000.000

Tenga presente que los datos de ambas tablas las deben crecer proporcionalmente, según los requerimientos de la aplicación.

Caso 2: Evalúe el join cuando la tabla con más registros tenga más de 10.000.000 de tuplas.

10.00.000 < Tr < inf.

Tenga presente que los datos de ambas tablas las deben crecer proporcionalmente, según los requerimientos de la aplicación.

El valor de M permanece constante para los tres casos.

Haga un cuadro resumen de las estrategias para cada caso e indique cuál estrategia debería seleccionar el optimizador en cada caso.

Indique cual tabla asume como R y cual como S. Además especificar las imágenes de los índices.

Caso X:

|  |  |  |  |
| --- | --- | --- | --- |
| **Datos de entrada por tabla** | **Estrategia Usada** | **Fórmula usada para costo de la entrada** | **Costo entrada.** |
| Esquema de almacenamiento usado por R y por S  Br= Bs= Ib.s  Tr= Ts= Ib.r  M= | Usando índice en tabla 1 |  |  |
| usando índice en tabla B |  |  |
| usando ambos índices |  |  |

NOTAS:

Si sus tablas tienen menos registros de los indicados asuma los valores que en este punto se establecen. Recuerde que la base de datos debe crecer entonces en todas sus tablas de una manera proporcional.

**II. Optimización algebraica (30%)**

1. Defina una vista SQL que involucre tres tablas de su base de datos..

(La vista no debe contener ORDER BY ni GROUP BY)

2. Defina una consulta sobre la vista que acabo de crear. **La consulta de usuario debe contener ORDER BY o GROUP BY.**

3. Problema: Encontrar la expresión correspondiente del álgebra y optimizarla usando el algoritmo visto en clase. Tenga presente que el algoritmo visto en clase no contemplaba que hacer con el ORDER BY ni con el GROUP BY. Usted debe sugerir una variante al algoritmo para tratar esta situación.