

Fragmentación Horizontal Primaria en Bases de Datos Relacionales

Karla Guadalupe Robert Roa
Bases De Datos Distribuidas

La fragmentación es el proceso encargado de dividir una relación en otras subrelaciones de menor tamaño, y su objetivo es encontrar la unidad apropiada de distribución.

Un fragmento horizontal de una relación es un subconjunto de las tuplas de la relación. Las tuplas que pertenecen al fragmento horizontal se especifican mediante una condición sobre uno o más de los atributos de la relación, normalmente sólo interviene un atributo.

La fragmentación horizontal divide la relación horizontalmente agrupando filas para crear subconjuntos de tuplas, donde cada subconjunto tiene cierto significado lógico y éstos pueden asignarse a diferentes sitios en el sistema distribuido.

La Fragmentación Horizontal Primaria: Se define como una operación de selección (σ) sobre una relación del esquema de la base de datos: Dada una relación R, un fragmento horizontal primario sería $\sigma_{\text{Predicado}}(R)$.

El objetivo es mejorar el rendimiento de las consultas y la gestión de los datos, adaptando la estructura a las necesidades reales de uso.

Algoritmo COM_MIN propuesto por Tamer Özsu.

Este algoritmo se basa en la idea de que las consultas a la base de datos utilizan un conjunto de predicados simples (condiciones) que se pueden combinar para definir los fragmentos. Estos predicados se usan para definir fragmentos horizontales basados en minterms.

Paso 1: Identificación de la relación y los predicados

Primero, se elige la relación (tabla) que se va a fragmentar, identificando los predicados de selección que representan las consultas más frecuentes. Esto permite alinear los fragmentos con el acceso real de los usuarios.

Ejemplo práctico:

Empleados(Nombre, Edad, Departamento)

Predicados de selección típicos:

- ♥ P1: Edad > 40
- ♥ P2: Departamento = 'Ventas'

Paso 2: Construcción de minterms

Los minterms son combinaciones de los predicados y sus negaciones, generando un conjunto de condiciones que cubren todas las posibilidades lógicas. Cada minterm define un fragmento único.

- ♥ m1: $\text{Edad} > 40 \wedge \text{Departamento} = \text{'Ventas'}$
- ♥ m2: $\text{Edad} > 40 \wedge \text{Departamento} \neq \text{'Ventas'}$
- ♥ m3: $\text{Edad} \leq 40 \wedge \text{Departamento} = \text{'Ventas'}$
- ♥ m4: $\text{Edad} \leq 40 \wedge \text{Departamento} \neq \text{'Ventas'}$

Los minterms permiten que cada tupla pertenezca exactamente a un fragmento, asegurando la no duplicidad.

Paso 3: Evaluación de los minterms

Antes de crear físicamente los fragmentos, es necesario verificar que los minterms generen fragmentos no vacíos. Si un minterm no corresponde a ningún registro, se descarta. Esto evita fragmentos innecesarios y optimiza el espacio.

Por ejemplo, si no hay empleados menores o iguales a 40 años en el departamento de Ventas, el minterm 3 generaría un fragmento vacío.

Paso 4: Creación de fragmentos

Cada minterm verificado genera un fragmento horizontal. Estos fragmentos se representan como consultas SQL que extraen los registros que cumplen las condiciones de cada minterm.

Fragmento 1:

```
SELECT * FROM Empleados WHERE Edad > 40 AND Departamento = 'Ventas';
```

Fragmento 2:

```
SELECT * FROM Empleados WHERE Edad > 40 AND Departamento <> 'Ventas';
```

Fragmento 3:

```
SELECT * FROM Empleados WHERE Edad <= 40 AND Departamento = 'Ventas';
```

Fragmento 4:

```
SELECT * FROM Empleados WHERE Edad <= 40 AND Departamento <> 'Ventas';
```

Cada fragmento se almacena como una vista o tabla parcial, dependiendo del sistema de gestión de base de datos utilizado. Esto permite consultas más rápidas, ya que cada fragmento es más pequeño y contiene solo los datos relevantes para ciertos criterios.

Ventajas del Algoritmo COM_MIN

- ♥ **Garantiza fragmentos disjuntos:** cada tupla está en un único fragmento, sin solapamientos.
- ♥ **Mejora el rendimiento:** las consultas se ejecutan sobre fragmentos más pequeños, acelerando las búsquedas.
- ♥ **Optimiza recursos:** solo se crean los fragmentos necesarios, reduciendo el almacenamiento.
- ♥ **Adaptable:** el conjunto de fragmentos se basa en las consultas reales, adaptándose a las necesidades de uso.

Referencia

Özsu, M. T., & Valduriez, P. (2011). *Principles of Distributed Database Systems* (3rd ed.). Springer.