



**INSTITUTO POLITÉCNICO NACIONAL**



**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

**UPIITA**

**PROFESOR:** Carlos De La Cruz Sosa

**ASIGNATURA:** Bases de Datos Distribuidas

**GRUPO:** 3TM3

**Tarea: Fragmentacion horizontal primaria en base  
de datos relacionales**

**EQUIPO 5:**

- Bernal Aguilar Yuvia Abigail
- Contreras Jimenez Mariana Montserrat
- Medina Gómez Jimena Zarahí

## Fragmentacion horizontal primaria en base de datos relacionales

Se le pidio a ChatGpt lo siguiente:

“ ¿Comó diseñar fragmentos horizontales primarios en datos de bases relacionales basados en la teoria de tamer ozsu usando el algoritmo com\_min. Consideraciones para fragmentar “

Y nos arrojo lo siguiente:

El diseño de fragmentos horizontales primarios basado en la teoría de Tamer Özsu y el algoritmo COM\_MIN es un enfoque sistemático para dividir relaciones en fragmentos basados en predicados de selección. Aquí te explico el proceso:

### Fundamentos Teóricos

Los fragmentos horizontales primarios son subconjuntos de tuplas de una relación definidos por predicados mínimos y completos. Tamer Özsu propone que estos fragmentos deben cumplir con:

1. Completitud: Cada tupla de la relación original debe pertenecer a algún fragmento
2. Reconstrucción: La relación original debe poder reconstruirse mediante la unión de todos los fragmentos
3. Disyunción: Los fragmentos deben ser disjuntos (idealmente) para minimizar la redundancia

### Algoritmo COM\_MIN

El algoritmo COM\_MIN se utiliza para encontrar un conjunto mínimo y completo de predicados para la fragmentación:

1. Entrada: Conjunto de predicados simples  $P = \{p_1, p_2, \dots, p_n\}$
2. Salida: Conjunto mínimo completo  $M = \{m_1, m_2, \dots, m_k\}$

Pasos del Algoritmo:

1. Inicializar  $M = \emptyset$
2. Ordenar los predicados en  $P$  por selectividad (de menos a más selectivos)
3. Para cada predicado  $p$  en  $P$ :
  - Si  $p$  no está cubierto por ningún predicado en  $M$ :
    - Añadir  $p$  a  $M$
    - Eliminar de  $P$  todos los predicados cubiertos por  $p$
4. Repetir hasta que  $P$  esté vacío

### Consideraciones para Fragmentar

#### 1. Análisis de Aplicaciones

- Identificar las consultas frecuentes y sus predicados de selección
- Analizar las operaciones de actualización y su impacto en los fragmentos

## 2. Selección de Predicados

- Los predicados deben ser relevantes para las aplicaciones
- Deben tener suficiente selectividad para justificar la fragmentación
- Evitar predicados que generen fragmentos muy pequeños o muy grandes

## 3. Evaluación de Fragmentos

- Verificar que el conjunto de predicados sea completo (cubre todas las tuplas)
- Asegurar que sea mínimo (sin redundancias)
- Calcular el tamaño estimado de cada fragmento

## 4. Optimización

- Considerar el balance de carga entre fragmentos
- Evaluar el costo de acceso a los fragmentos
- Analizar el impacto en las operaciones distribuidas (joins, unions)

## Ejemplo Práctico

Para una relación EMPLEADO(emp\_id, nombre, depto, salario, ciudad):

1. Predicados identificados:
  - P1: depto = "Ventas"
  - P2: depto = "TI"
  - P3: salario > 5000
  - P4: ciudad = "Madrid"
2. Aplicar COM\_MIN:
  - Suponiendo que P1 y P2 cubren todos los departamentos, y son mutuamente excluyentes
  - $M = \{P1, P2\}$  (si no hay otros predicados relevantes)
3. Fragmentos resultantes:
  - F1:  $\sigma(\text{depto}=\text{"Ventas"})\text{(EMPLEADO)}$
  - F2:  $\sigma(\text{depto}=\text{"TI"})\text{(EMPLEADO)}$
  - F3:  $\sigma(\text{depto} \neq \text{"Ventas"} \wedge \text{depto} \neq \text{"TI"})\text{(EMPLEADO)}$

## Herramientas y Validación

- Utilizar matrices de acceso para validar la bondad de la fragmentación
- Calcular medidas de calidad como:
  - Localidad de acceso
  - Paralelismo potencial
  - Balance de carga

Este enfoque sistemático asegura que los fragmentos horizontales primarios estén optimizados para el entorno distribuido, mejorando el rendimiento mientras se mantienen las propiedades de integridad de los datos.

## **CONCLUSIONES.**

Medina Gómez Jimena Sarahí.

La fragmentación horizontal primaria, según la teoría de Tamer Özsu, permite organizar los datos en fragmentos lógicos y manejables, garantizando que cada tupla sea incluida sin duplicidad y que la relación original pueda ser reconstruida. Esta estructura mejora notablemente el rendimiento en entornos distribuidos al reducir el volumen de datos que cada nodo necesita manejar.

Contreras Jiménez Mariana Montserrat.

El algoritmo COM\_MIN es clave para diseñar fragmentos eficientes, ya que selecciona un conjunto mínimo y completo de predicados, eliminando redundancias y asegurando que cada fragmento tenga un propósito específico. Esta precisión en la selección permite adaptar la base de datos a las necesidades reales de consulta, mejorando la velocidad de acceso y la escalabilidad del sistema.

Berna Aguilarl Yuvia Abigail.

Fragmentar con criterio no solo implica aplicar una técnica matemática, sino también comprender el comportamiento del sistema: las consultas más frecuentes, la carga de trabajo y el tamaño de los fragmentos deben considerarse para lograr una distribución equilibrada. Esto permite lograr mayor paralelismo, optimizar recursos y construir una base sólida para operaciones distribuidas sin comprometer la integridad de los datos.

## **CONCLUSIÓN GENERAL.**

El diseño de fragmentos horizontales primarios en bases de datos relacionales, fundamentado en la teoría de Tamer Özsu, permite dividir las relaciones en subconjuntos lógicos que mejoran el rendimiento y organización de los datos, garantizando siempre la posibilidad de reconstrucción y la no duplicidad de registros. El uso del algoritmo COM\_MIN es esencial para lograr una fragmentación óptima, ya que selecciona un conjunto mínimo y completo de predicados, evitando redundancias y adaptando la estructura de los datos a las consultas más relevantes. No obstante, una fragmentación efectiva va más allá del enfoque teórico: requiere un análisis profundo del comportamiento del sistema, considerando la carga de trabajo, la frecuencia de acceso y el equilibrio entre fragmentos para lograr una distribución eficiente, escalable y alineada con las necesidades reales de operación en entornos distribuidos.