INSTITUTO POLITECNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLGIAS AVANZADAS - IPN

MATERIA

Bases de Datos Distribuidas

PROFESOR

Carlos De La Cruz Sosa

ALUMNOS

Fernández Guerrero Keb Sebastián

Ramírez Orozco Juan Carlos

Sánchez Herrera Armando Eduardo

TEMA

Investigación sobre Fragmentación Horizontal Primaria y Derivada

Grupo 3TM3 Equipo 06 Tarea No. 02

Fragmentación Horizontal Primaria

La fragmentación horizontal primaria (FHP) es una técnica de diseño en bases de datos distribuidas que divide una relación (tabla) en *subconjuntos de tuplas* basados en un predicado definido sobre los atributos de la relación misma. Cada fragmento resultante contiene un grupo de filas que satisfacen una condición específica, usualmente vinculada a las consultas frecuentes o a la localidad de los datos (Özsu & Valduriez, 2020).

Características clave según Özsu:

1. **Predicados de fragmentación**: Se definen condiciones (ej: `WHERE sucursal = "Norte"`) para crear fragmentos.
2. **Completitud**: Cada tupla de la relación original debe pertenecer a algún fragmento.

3. **Reconstrucción**: La unión de todos los fragmentos debe regenerar la relación original sin pérdida de datos.

4. **Disyunción mínima**: Idealmente, los fragmentos no deben superponerse (aunque en la práctica esto depende del diseño).

Ejemplo:

Si tenemos una tabla `EMPLEADOS` y la fragmentamos por región:

- `Fragmento_Emp_Norte`: `WHERE region = 'Norte'`

- `Fragmento_Emp_Sur`: `WHERE region = 'Sur'`

Algoritmos de Derivación de Predicados

La FHP requiere definir predicados lógicos para dividir una tabla. Özsu & Valduriez (2020) describen métodos sistemáticos para derivarlos, especialmente en entornos donde las consultas son complejas o involucran múltiples atributos.

Algoritmo Básico (de Minería de Predicados)

1. Recolección de predicados candidatos:

- Analizar las condiciones (`WHERE`) de las consultas frecuentes.

- Ejemplo: Si las consultas filtran por `departamento = "Ventas"` y `sucursal = "Norte"` estos son predicados candidatos.

2. Minimización del conjunto de predicados:

- Eliminar redundancias (ej: `sucursal = "Norte" AND sucursal = "Norte"`).

- Combinar predicados compatibles (ej: `departamento = "Ventas" OR departamento = "Marketing"` → `departamento IN ("Ventas", "Marketing")`).

3. Generación de predicados mínimos (Algoritmo de Compleción):

- Asegurar que los predicados sean *completos* (cubran todas las tuplas) y *disjuntos* (no se superpongan, salvo en FHP con superposición controlada).

Ejemplo Práctico:

Para una tabla `PEDIDOS` con consultas frecuentes como:

- `WHERE cliente_region = "Este" AND año = 2023`

- `WHERE cliente_region = "Oeste"`

Predicados derivados:

- `P1: cliente_region = "Este" AND año = 2023`

- `P2: cliente_region = "Oeste"`

- `P3: NOT (P1 OR P2)` (fragmento residual para completitud).

Esquemas de Fragmentación Horizontal Primaria

Özsu distingue entre dos enfoques principales:

a) Fragmentación Primaria Directa

- *Definición*: Se aplica directamente a la relación original usando predicados locales.

- *Condiciones:*

  - Los predicados deben ser *relevantes para las consultas* de un sitio específico.

  - Ejemplo: Fragmentar `EMPLEADOS` por `sucursal_id = X` si cada sitio accede principalmente a su propia sucursal.

b) Fragmentación Primaria Derivada

- *Definición*: La fragmentación se basa en predicados de *otra relación relacionada* (vía joins).

- *Ejemplo*:

  - Si `DEPARTAMENTOS` está fragmentada por `region`, entonces `EMPLEADOS` puede fragmentarse derivadamente con:

```
WHERE EMPLEADOS.depto_id IN (SELECT id FROM DEPARTAMENTOS WHERE region = "Norte")
```

Reglas Clave:

- *Completitud*: Todos los datos deben estar asignados a algún fragmento (Özsu & Valduriez, 2020).

- *Reconstrucción*: La unión de fragmentos debe igualar la tabla original.

- *Optimización*: Los fragmentos deben minimizar accesos remotos (ej: almacenar datos cercanos a donde se consultan).

Ejemplo Práctico con SQL: Fragmentación Horizontal Primaria

Contexto:

Tenemos una tabla `EMPLEADOS` en una empresa multiregional, con consultas frecuentes filtradas por `region` y `departamento`.

Paso 1: Definir predicados de fragmentación

Analizando las consultas, identificamos:

- Sitio Norte: `WHERE region = 'Norte' AND departamento IN ('Ventas', 'Logística')`.

- Sitio Sur: `WHERE region = 'Sur' AND departamento = 'TI'`.

Paso 2: Crear fragmentos en SQL

```sql
-- Fragmento para el Norte
CREATE TABLE empleados_norte AS
SELECT * FROM empleados
WHERE region = 'Norte' AND departamento IN ('Ventas', 'Logística');

-- Fragmento para el Sur
CREATE TABLE empleados_sur AS
SELECT * FROM empleados
WHERE region = 'Sur' AND departamento = 'TI';

-- Fragmento residual (opcional, para completitud)
CREATE TABLE empleados_resto AS
SELECT * FROM empleados
WHERE NOT (region = 'Norte' AND departamento IN ('Ventas', 'Logística'))
AND NOT (region = 'Sur' AND departamento = 'TI');
```

Paso 3: Verificar propiedades

- *Completitud*: La unión de los fragmentos reconstruye la tabla original.

- *Disyunción*: Los fragmentos no se superponen (si el diseño es correcto).

Comparación: Fragmentación Primaria (FHP) vs. Derivada (FHD)

Aspecto	**Fragmentación Horizontal Primaria (FHP)**	**Fragmentación Horizontal Derivada (FHD)**
Base de la división	Predicados sobre atributos de la tabla misma.	Predicados basados en una tabla relacionada (vía joins).
Ejemplo	WHERE region = 'Norte' (en EMPLEADOS).	WHERE empleado.depto_id IN (SELECT id FROM departamentos WHERE region = 'Norte').

Aspecto	Fragmentación Horizontal Primaria (FHP)	Fragmentación Horizontal Derivada (FHD)
Complejidad	Más simple, directa.	Más compleja, requiere análisis de joins.
Uso típico	Cuando los datos se filtran por atributos locales.	Cuando la fragmentación depende de relaciones entre tablas.
Reconstrucción	Unión de fragmentos.	Unión + joins con la tabla de referencia.
Ventaja	Rendimiento rápido para consultas locales.	Coherencia con datos distribuidos en tablas relacionadas.

Ejemplo de FHD:

Fragmentar `PEDIDOS` basándose en la fragmentación de `CLIENTES`:

```sql
-- Asumiendo que CLIENTES está fragmentada por región
CREATE TABLE pedidos_este AS
SELECT p.* FROM pedidos p
JOIN clientes c ON p.cliente_id = c.id
WHERE c.region = 'Este';
```

Conclusión

- **FHP**: Ideal para datos independientes con acceso localizado.

- **FHD**: Útil cuando la distribución sigue jerarquías (ej: pedidos de clientes por región).

Fuentes Citadas (APA)

- Özsu, M. T., & Valduriez, P. (2020). *Principles of distributed database systems* (4th ed.). Springer.

- Ceri, S., & Pelagatti, G. (1984). *Distributed databases: Principles and systems*. McGraw-Hill.

Citado en el texto:

- Para FHP: (Özsu & Valduriez, 2020, Cap. 3).

- Para FHD: (Ceri & Pelagatti, 1984, p. 115).