EQUIPO2-TAREA 4

Fragmentación Horizontal Primaria y Derivada: Un Estudio de Caso sobre la Base de Datos AdventureWorks basado en los Principios de Özsu.

1.Análisis del Esquema Relacional Objetivo

Se realiza una revisión detallada de las tablas involucradas, basándose en la documentación del esquema de AdventureWorks.[11]

- **Production.ProductCategory:** Representa la agrupación de productos de más alto nivel. Contiene un número pequeño y fijo de categorías (4 filas: Bikes, Components, Clothing, Accessories) con ProductCategoryID como clave primaria.[16]
- **Production.ProductSubcategory:** Es la tabla objetivo para la fragmentación primaria. Se vincula a ProductCategory a través de la clave foránea ProductCategoryID, estableciendo una clara relación jerárquica.[12]
- **Production.Product:** La tabla principal de productos, vinculada a ProductSubcategory por ProductSubcategoryID. En la terminología de Özsu, esta es una relación "miembro" en una relación de tipo "propietario-miembro" con ProductSubcategory.[14]
- **Sales.SalesOrderDetail:** La tabla de detalles de transacciones, vinculada a Product a través de ProductID. Es una relación "miembro" con Product como "propietario".[17]

2. Deconstrucción de las Consultas de Alta Frecuencia

Se realiza un desglose riguroso de cada consulta para extraer su lógica de acceso a datos subyacente. Este es un tipo de análisis cuantitativo donde las consultas actúan como un proxy representativo de la carga de trabajo completa de la aplicación.[7]

- **Consulta (a): "Listar el producto más vendido de cada una de las categorías registradas en la base de datos."**
  - **Ruta de Acceso:** SalesOrderDetail → Product → ProductSubcategory.
  - **Operación Central:** GROUP BY ProductCategoryID.
  - **Implicación para la Fragmentación:** El rendimiento de esta consulta está directamente ligado a la eficiencia con la que el sistema puede aislar los

datos de cada categoría de producto. Proporciona una justificación fuerte y explícita para particionar los datos basándose en el atributo ProductCategoryID. Este es un ejemplo clásico de cómo una aplicación impulsa el diseño de la fragmentación.[22]

- **Consulta (b): "Listar el nombre de los clientes con más ordenes por cada uno de los territorios registrados en la base de datos."**
  - **Ruta de Acceso:** SalesOrderHeader ⟶ Customer ⟶ SalesTerritory.
  - **Operación Central:** GROUP BY TerritoryID.
  - **Implicación para la Fragmentación:** Esta consulta opera en una parte completamente diferente del esquema (el dominio de ventas/clientes) y particiona los datos basándose en un atributo geográfico (TerritoryID).[23] Por lo tanto, es irrelevante para la fragmentación de las tablas relacionadas con productos y no se utilizará para definir los fragmentos en este análisis.
- **Consulta (c): "Listar los datos generales de las ordenes que tengan al menos los mismos productos de la orden con salesorderid = 43676."**
  - **Ruta de Acceso:** Implica JOINs complejos o subconsultas sobre la propia tabla SalesOrderDetail.
  - **Operación Central:** Comparación basada en el contenido de conjuntos de productos, no una evaluación de predicados simples.
  - **Implicación para la Fragmentación:** Esta consulta no produce un predicado simple de la forma Atributo θ Valor que se requiere para la FHP.[3] Su rendimiento se
    beneficiará de la fragmentación derivada de la Consulta (a) (ya que operará en fragmentos de SalesOrderDetail más pequeños y pre-particionados), pero no puede ser utilizada para *definir* dichos fragmentos.

Paso 1: Definición del Conjunto Inicial de Predicados Simples (Pr)

Basado en el análisis de la Parte II, la Consulta (a) es la única consulta relevante para fragmentar la tabla ProductSubcategory. El atributo de partición identificado es ProductCategoryID. Para formar los predicados simples, necesitamos los valores distintos de este atributo. La tabla Production.ProductCategory en la base de datos AdventureWorks contiene cuatro categorías.[16] Asumiremos que sus IDs son 1, 2, 3 y 4, que corresponden a "Bikes", "Components", "Clothing" y "Accessories", respectivamente.

Por lo tanto, el conjunto inicial de predicados simples Pr para la relación ProductSubcategory es:

- p1:ProductCategoryID=1
- p2:ProductCategoryID=2
- p3:ProductCategoryID=3
- p4:ProductCategoryID=4
-

Construcción de Predicados Minterm

Un predicado minterm es una conjunción (una operación AND lógica) de predicados simples del conjunto Pr'.[3] Para un conjunto de

k predicados simples, existen 2k minterms potenciales. El propósito de generar minterms es asegurar que los fragmentos resultantes sean mutuamente excluyentes (disjuntos).

En nuestro caso, el conjunto de predicados Pr' es:
{ProductCategoryID = 1, ProductCategoryID = 2, ProductCategoryID = 3, ProductCategoryID = 4}
Estos predicados son inherentemente mutuamente excluyentes, ya que un ProductCategoryID no puede tener dos valores diferentes simultáneamente. Por ejemplo, la conjunción (ProductCategoryID = 1) AND (ProductCategoryID = 2) es una contradicción lógica y siempre evaluará a falso, resultando en un fragmento vacío. De manera similar, una conjunción como (ProductCategoryID = 1) AND NOT(ProductCategoryID = 1) también es una contradicción.

Debido a esta exclusividad mutua, los únicos predicados minterm significativos y no vacíos son los propios predicados simples. El proceso formal de generar todos los 24=16 minterms y luego eliminar los contradictorios se simplifica a usar directamente nuestros predicados originales.

Por lo tanto, el conjunto de predicados minterm M es:

- m1:ProductCategoryID=1
- m2:ProductCategoryID=2
- m3:ProductCategoryID=3
- m4:ProductCategoryID=4

4.2 Definición de los Fragmentos Horizontales Primarios

Un fragmento horizontal Ri se define aplicando su predicado minterm correspondiente mi a la relación base R, utilizando el operador de selección del álgebra relacional: $R_i = \sigma_{m_i}(R)$.[3]

A continuación, se definen los fragmentos para Production.ProductSubcategory en el formato descriptivo solicitado, junto con los conjuntos de predicados que los originan.

Conjunto PR' (Predicados Simples, Completos y Mínimos)

```
PR'_ProductSubCategory = {
  p1: ProductCategoryID = 1,
  p2: ProductCategoryID = 2,
  p3: ProductCategoryID = 3,
  p4: ProductCategoryID = 4
}
```

Conjunto M (Predicados Minterm)

```
M_ProductSubCategory = {
  m1: ProductCategoryID = 1,
  m2: ProductCategoryID = 2,
  m3: ProductCategoryID = 3,
  m4: ProductCategoryID = 4
}
```

Diseño del Esquema de Fragmentación Derivada

Esta sección extiende el diseño a las tablas dependientes, Product y SalesOrderDetail, utilizando la Fragmentación Horizontal Derivada (FHD). Se comienza justificando explícitamente por qué la FHD es el enfoque apropiado.

5.1 Justificación de la Fragmentación Derivada vs. Primaria

Es crucial entender por qué no se aplica la fragmentación primaria a las tablas Product y SalesOrderDetail. Estas tablas son relaciones "miembro" en enlaces de equijoin con sus relaciones "propietario" (ProductSubcategory y Product, respectivamente).[9]

El principal beneficio de la FHD es preservar la localidad de los datos para las operaciones de JOIN. Al fragmentar una tabla miembro basándose en la fragmentación de su propietario, se asegura que los datos relacionados (por ejemplo, un producto y todos sus detalles de venta) estén físicamente coubicados en el mismo sitio. Esto mejora drásticamente el rendimiento de las consultas que unen estas tablas, como la Consulta (a), al evitar costosos JOINs remotos.[9] Intentar una fragmentación primaria en estas tablas usando sus propios atributos rompería esta coubicación, degradando el rendimiento. Por ejemplo, fragmentar

Product por Color no garantizaría que todos los productos de la categoría "Bikes" residan juntos.

5.2 Fragmentación Horizontal Derivada de Production.Product

La FHD se define formalmente utilizando el operador semijoin ($\ltimes$): $R_i = R \ltimes S_i$, donde R es la relación miembro y Si son los fragmentos de la relación propietario S.[9]

- **Relación Propietario:** ProductSubcategory (fragmentada en ProductSubcategory_1 a ProductSubcategory_4).
- **Relación Miembro:** Product.
- **Condición de JOIN:** Product.ProductSubcategoryID IN (SELECT ProductSubcategoryID FROM ProductSubcategory).
- **Regla de Derivación:** Los fragmentos de Product se definen mediante un semijoin con los fragmentos de ProductSubcategory.
    - Product_1 = Product $\ltimes$ ProductSubcategory_1
    - Product_2 = Product $\ltimes$ ProductSubcategory_2

    - Product_3 = Product ⋉ ProductSubcategory_3
    - Product_4 = Product ⋉ ProductSubcategory_4
- **Resultado:** Se crean cuatro fragmentos de la tabla Product. Product_1 contiene todos los productos de la categoría "Bikes", Product_2 todos los de "Components", y así sucesivamente.

5.3 Fragmentación Horizontal Derivada de Sales.SalesOrderDetail

El proceso se repite, propagando la fragmentación hacia abajo en la jerarquía del esquema. Esta propagación demuestra el "efecto dominó" de la fragmentación: la decisión de una clave de fragmentación primaria tiene un impacto en cascada a través del esquema.

- **Relación Propietario:** Product (ahora fragmentada en Product_1 a Product_4).
- **Relación Miembro:** SalesOrderDetail.
- **Condición de JOIN:** SalesOrderDetail.ProductID = Product.ProductID.
- **Regla de Derivación:** Los fragmentos de SalesOrderDetail se definen mediante un semijoin con los fragmentos recién creados de Product.
    - SalesOrderDetail_1 = SalesOrderDetail ⋉ Product_1
    - SalesOrderDetail_2 = SalesOrderDetail ⋉ Product_2
    - SalesOrderDetail_3 = SalesOrderDetail ⋉ Product_3
    - SalesOrderDetail_4 = SalesOrderDetail ⋉ Product_4
- **Resultado:** Se crean cuatro fragmentos de la tabla SalesOrderDetail. SalesOrderDetail_1 contendrá todas las líneas de pedido de productos de la categoría "Bikes", SalesOrderDetail_2 todas las de "Components", etc. Esto asegura que cuando se ejecute la Consulta (a) para la categoría "Bikes", todas las tablas necesarias (ProductSubcategory_1, Product_1, SalesOrderDetail_1) puedan estar coubicadas, maximizando el rendimiento.

La siguiente tabla proporciona un resumen conciso y claro de las reglas de fragmentación derivada, mostrando la propagación de la decisión de fragmentación inicial a través del esquema.

6.1 El Esquema de Fragmentación Completo

El diseño final resulta en un esquema donde las tres tablas relacionadas con productos (ProductSubcategory, Product y SalesOrderDetail) están particionadas horizontalmente en cuatro fragmentos cada una. La partición se basa en el atributo ProductCategoryID, que se propaga desde la tabla primaria (ProductSubcategory) a las tablas derivadas.

- **Fragmentos de ProductSubcategory (Primarios):**
  - ProductSubcategory_1 (Categoría 1: Bikes)
  - ProductSubcategory_2 (Categoría 2: Components)
  - ProductSubcategory_3 (Categoría 3: Clothing)
  - ProductSubcategory_4 (Categoría 4: Accessories)
- **Fragmentos de Product (Derivados):**
  - Product_1 (Productos de la Categoría 1)
  - Product_2 (Productos de la Categoría 2)
  - Product_3 (Productos de la Categoría 3)
  - Product_4 (Productos de la Categoría 4)
- **Fragmentos de SalesOrderDetail (Derivados):**
  - SalesOrderDetail_1 (Líneas de pedido de la Categoría 1)
  - SalesOrderDetail_2 (Líneas de pedido de la Categoría 2)
  - SalesOrderDetail_3 (Líneas de pedido de la Categoría 3)
  - SalesOrderDetail_4 (Líneas de pedido de la Categoría 4)