

# Fragmentación Horizontal en AdventureWorks

Basado en Tamer Özsu y AdventureWorks

25 de junio de 2025

## 1. Fundamentos Teóricos

Según Tamer Özsu, la *fragmentación horizontal primaria* se basa en particionar una relación en función de predicados de selección sobre sus atributos, de modo que:

- Cada fragmento sea *disjunto* (no haya tuplas repetidas).
- La unión de todos los fragmentos sea *completa* (no se pierda información).
- Los predicados sean *mínimos* (no incluyan condiciones redundantes).

La *fragmentación derivada* aplica una especie de semijoin para propagar la partición a tablas relacionadas (por clave foránea), garantizando coherencia entre fragmentos relacionados.

## 2. Fragmentación Horizontal Primaria sobre Production.ProductSubCategory

### 2.1. Identificación de predicados candidatos

A partir de las consultas de más alta frecuencia se detecta que `productcategoryid` es el atributo clave. Definimos el conjunto de predicados PR (uno por cada categoría existente) `:contentReference[oaicite:1]index=1`:

$$PR_{ProductSubCategory} = \{ P_1 : productcategoryid = 1, \quad P_2 : productcategoryid = 2, \\ P_3 : productcategoryid = 3, \quad P_4 : productcategoryid = 4 \}.$$

Tras descartar la combinación lógica de todos (no aporta fragmento útil), obtenemos el conjunto mínimo y completo:

$$M_{ProductSubCategory} = \{ P_1, P_2, P_3, P_4 \}.$$

## 2.2. Creación de fragmentos primarios

Para cada predicado generamos un fragmento usando `SELECT ... INTO`:

```
1  -- Fragmento FPSC1: productcategoryid = 1
2  SELECT *
3      INTO FPSC1
4      FROM AdventureWorks.Production.ProductSubCategory
5      WHERE productcategoryid = 1;
6
7  -- Fragmento FPSC2: productcategoryid = 2
8  SELECT *
9      INTO FPSC2
10     FROM AdventureWorks.Production.ProductSubCategory
11     WHERE productcategoryid = 2;
12
13 -- Fragmento FPSC3: productcategoryid = 3
14 SELECT *
15     INTO FPSC3
16     FROM AdventureWorks.Production.ProductSubCategory
17     WHERE productcategoryid = 3;
18
19 -- Fragmento FPSC4: productcategoryid = 4
20 SELECT *
21     INTO FPSC4
22     FROM AdventureWorks.Production.ProductSubCategory
23     WHERE productcategoryid = 4;
```

Cada uno de estos fragmentos es disjunto y, al unirlos, recuperamos la tabla original.

## 3. Fragmentación Derivada

### 3.1. Sobre Production.Product

Aplicamos un semijoin para derivar la partición de `ProductSubCategory`:

```
1  -- Derivar FP1 asociado a FPSC1
2  SELECT P.*
3      INTO FP1
4      FROM AdventureWorks.Production.Product AS P
5      JOIN FPSC1
6          ON P.productsubcategoryid = FPSC1.productsubcategoryid;
7
8  -- Derivar FP2 asociado a FPSC2
9  SELECT P.*
10     INTO FP2
11     FROM AdventureWorks.Production.Product AS P
12     JOIN FPSC2
13         ON P.productsubcategoryid = FPSC2.productsubcategoryid;
```

```
14
15 -- (y as sucesivamente para FPSC3, FPSC4)
```

### 3.2. Sobre Sales.SalesOrderDetail

Propagamos la partición a los detalles de venta:

```
1 -- Derivar FSOD1 asociado a FP1
2 SELECT SOD.*
3 INTO FSOD1
4 FROM AdventureWorks.Sales.SalesOrderDetail AS SOD
5 JOIN FP1
6 ON SOD.productid = FP1.productid;
7
8 -- Derivar FSOD2 asociado a FP2
9 SELECT SOD.*
10 INTO FSOD2
11 FROM AdventureWorks.Sales.SalesOrderDetail AS SOD
12 JOIN FP2
13 ON SOD.productid = FP2.productid;
14
15 -- (y as sucesivamente para FP3, FP4)
```

De esta forma garantizamos que cada fragmento de detalle de venta sólo contiene órdenes con productos de la subcategoría correspondiente.

## 4. Explicación de cada paso

1. **Selección de predicados:** Se elige el atributo más utilizado en las consultas frecuentes (`productcategoryid`).
2. **Generación de PR:** Un predicado por valor distinto de `productcategoryid`.
3. **Filtrado a Mínimo y Completo:** Se descartan combinaciones redundantes, quedando un fragmento por cada valor.
4. **Creación de fragmentos primarios:** Con `SELECT ... INTO` creamos tablas físicas disjuntas.
5. **Fragmentación derivada:** Con `JOIN` propagamos cada fragmento a tablas relacionadas, conservando sólo las tuplas relevantes.
6. **Reconstrucción:** La unión de todos los fragmentos primarios/derivados reproduce exactamente las tablas originales sin pérdida ni duplicación.