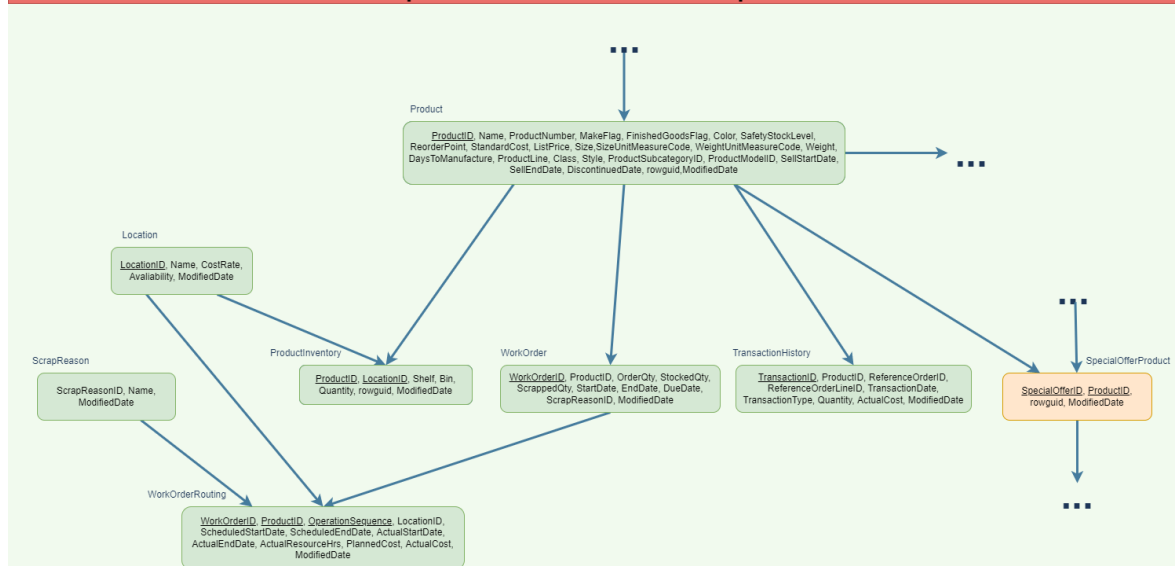


Diseño de fragmentación horizontal

Grafo de conceptual dada la tabla prioritaria: Product



Aplicando algoritmo COM_MIN

Nuestro equipo ha tomado los esquemas: Sales y Production proponiendo las siguientes 12 consultas:

- 1.- Listar todos los productos que tengan un precio de venta mayor o igual a 40 dólares.
- 2.- Listar la cantidad de productos que se venden para ensamblar en casa
- 3.- Listar los productos que les toma un día o más en ensamblar el producto
- 4.- Listar la cantidad de productos Amarillos
- 5.- Registrar una nueva orden de fabricación de 4 piezas de un mismo producto.
- 6.- Registrar una nueva transacción de tipo Venta
- 7.- Verificar que productos no cuentan con una oferta/descuento
- 8.- Listar la cantidad total de productos por el área en el que se encuentran dentro de la planta de producción.
- 9.- Comprobar que transacciones del tipo Venta han superado el costo del promedio de solo este tipo.
- 10.- Obtener la cantidad de productos que ya se encuentran en el área de ensamblaje final
- 11.- Listar las ordenes de producción que terminaron de fabricarse el 1 de enero del año 2013
- 12.- Listar las ordenes de manufactura que no cumplieron con su meta de stock

Obtención de los predicados por tablas:

PRProduct = { (Tabla Prioritaria)

P1: ListPrice >= 40

P3: MakeFlag = 1

P3: Color = 'Yellow'

P4: DaysToManufacture >= 1

}

PRWorkOrder = {

P5: OrderQty = 4

P6: EndDate = '2013-01-01'

P7: ScrappedQty > 0

}

PRSpecialOfferProduct = {

P8: SpecialOfferID = 1

}

PRTransactionHistory = {

P9: TransactionType = 'S'

}

PRProductInventory = {

P10: LocationID = 1

P11: LocationID = 2

P12: LocationID = 3

P13: LocationID = 4

P14: LocationID = 5

P15: LocationID = 6

P16: LocationID = 7

P17: LocationID = 10

P18: LocationID = 20

P19: LocationID = 30

P20: LocationID = 40

P21: LocationID = 45

P22: LocationID = 50

P23: LocationID = 60

}

Ahora aplicaremos en algoritmo COM_MIN sobre la tabla Propietaria Product y su conjunto PRProduct

1. Encontramos un Pi que particione a Product en 2 fragmentos:
(P3 no fue agregado por devolver una cantidad inequitativa de tuplas con relación a todos los registros)

F1: ListPrice >= 40 F2: ListPrice < 40 and MakeFlag <> 1 and DaysToManufacture < 1

F3: MakeFlag = 1

F4: DaysToManufacture >= 1

PR' = {

P1

P24: ListPrice < 40

P2

P25: MakeFlag = 0

P4

P26: DaysToManufacture < 1

}

Cuando hacemos los pasos iterativos hay que comprobar la relevancia entre los predicados:

$$\text{Acc(mi)} / \text{Card(fi)} = \text{Acc(mj)} / \text{Card(fj)}$$

- Acc(mi) – Frecuencia de acceso a los datos seleccionados por el minitermino i
Donde mi es la conjunción de predicados simples del conjunto PR considerando la forma natural y forma negada de los predicados
- Card(fi) – Cardinalidad del fragmento – el total de tuplas del fragmento

El conjunto de predicados miniterminos tiene 2^n predicados

M = {

M1: P1 and P2 and P4

M2: P1 and P2 and not P4

M3: P1 and not P2 and P4

M4: P1 and not P2 not P4

M5: not P1 and P2 P4

M6: not P1 and P2 not P4

M7: not P1 and not P2 and P4

M8: not P1 and not P2 and not P4

}

Acc(m1) = 0.333...

Card(f1) = 261

Acc(m2) = 0.333...

Card (f2) = 239

Acc(m3) = 0.333...

Card(f3) = 258