



**INSTITUTO POLITECNICO NACIONAL**  
**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y**  
**TECNOLOGIAS AVANZADAS**  
**INGENIERIA EN TELEMATICA**

**BASES DE DATOS DISTRIBUIDAS**

**EQUIPO 3:**

**ACOSTA VILLA CRISTIAN ABRAHAM**

**ARVIZU GÓMEZ ABRAHAM**

**ENRIQUEZ MELENDEZ JESÚS**

**GARCÍA ACOSTA ABRAHAM**

**MONZÓN MOGOLLAN DANIELA**

**ROSAS PALACIOS ALAN**

**PRESENTAN:**

**PROYECTO FINAL**

**ADVENTURE WORKS 2019**

**GRUPO:**

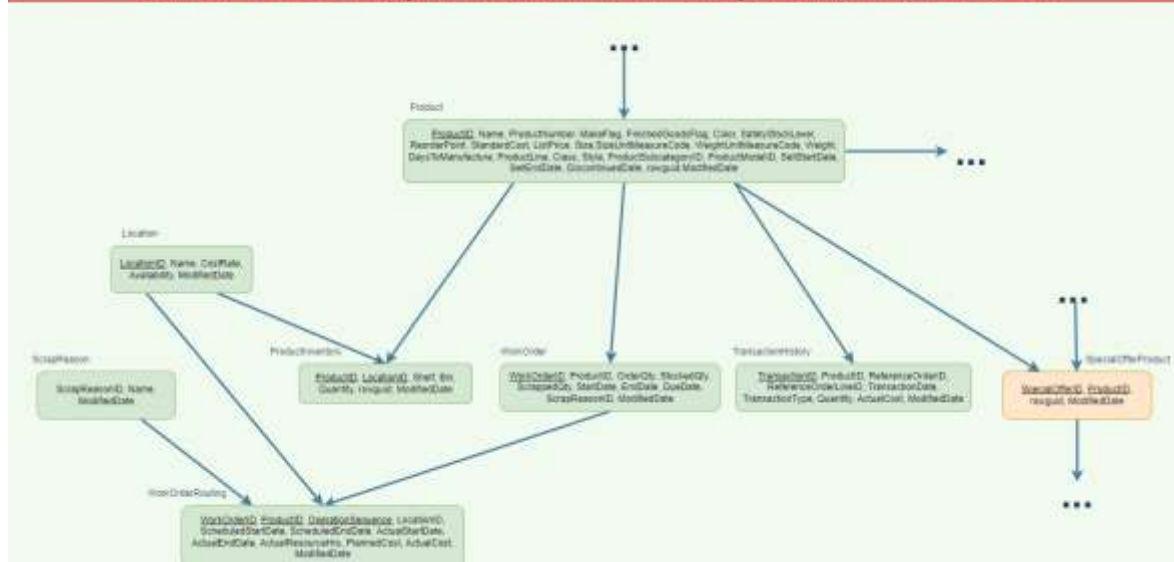
**3TV2**

**SEMESTRE 2022 – 1**



## 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

### Equipo 3 – Diseño de la base de datos distribuida

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. Verificamos la primera regla para aplicar el algoritmo COM\_MIN

PR' = {

}

No obtenemos ningún predicado en PR, debido a que solo se ocupa en 1 consulta para cada predicado.

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

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

- $\text{Acc}(mi)$  – 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
- $\text{Card}(fi)$  – Cardinalidad del fragmento – el total de tuplas del fragmento

El conjunto de predicados mini términos tiene  $2^n$  predicados

M = {

}

El problema con las consultas de la tabla propietaria y con las tablas derivadas es que todas son independientes y ninguna divide al bloque de consultas en 2 como lo especifica la regla #1 del algoritmo COM\_MIN, por lo cual de las consultas que tenemos presentes remplazaremos 3 para un esquema diferente más, que se relacionen con “Producción” y de esta manera poder fragmentar la base de datos (3 consultas para el esquema).

## Equipo 3 – Diseño de la base de datos distribuida

Los esquemas fueron alojados en una instancia de SQL Server diferente cada uno, esto para cumplir con la fragmentación por esquemas.

Al dividir la base de datos en esquemas, identificamos las tablas a utilizar para hacer una posterior implementación en table script. Las siguientes son las tablas que se utilizaron:

- PRProduct
- PRWorkOrder
- PRProductVendor
- PRPurchaseOrderDetail
- PRProductInventory

### **Integridad referencial**

Para cuidar la integridad referencial en los esquemas individuales se utilizó el concepto de llaves foráneas.

En cuanto a la integridad referencial cuando interactúan los dos esquemas notamos que no es necesario (debido a la estructura de nuestras operaciones de lectura y escritura, la implementación de disparadores dado que las operaciones DML son realizadas en un esquema cuya entidad referencial esta protegida por el concepto de llave foránea.

### **Operaciones complementarias de lectura y escritura**

Al momento de realizar las consultas no se tomaron en cuenta el mínimo de operaciones de lectura y escritura, es por esto que se agregaron dos instrucciones más de inserción las cuales se efectúan en las siguientes dos tablas:

- ProductInventory
- ScarpReason

Las nuevas operaciones son:

1. Agregar 666 registros de productos nuevos.
2. Registrar una nueva transacción tipo W de 5 piezas de un producto.